General parameter-shift rules for quantum gradients

David Wierichs[1,2], Josh Izaac[1], Cody Wang[3], and Cedric Yen-Yu Lin[3]

[1]Xanadu, Toronto, ON, M5G 2C8, Canada

[2]Institute for Theoretical Physics, University of Cologne, Germany

[3]AWS Quantum Technologies, Seattle, Washington 98170, USA

Variational quantum algorithms are ubiquitous in applications of noisy intermediate-scale quantum computers. Due to the structure of conventional parametrized quantum gates, the evaluated functions typically are finite Fourier series of the input parameters. In this work, we use this fact to derive new, general parameter-shift rules for single-parameter gates, and provide closed-form expressions to apply them. These rules are then extended to multi-parameter quantum gates by combining them with the stochastic parameter-shift rule. We perform a systematic analysis of quantum resource requirements for each rule, and show that a reduction in resources is possible for higher-order derivatives. Using the example of the quantum approximate optimization algorithm, we show that the generalized parameter-shift rule can reduce the number of circuit evaluations significantly when computing derivatives with respect to parameters that feed into many gates. Our approach additionally reproduces reconstructions of the evaluated function up to a chosen order, leading to known generalizations of the Rotosolve optimizer and new extensions of the quantum analytic descent optimization algorithm.

1 Introduction

With the advent of accessible, near-term quantum hardware, the ability to rapidly test and prototype quantum algorithms has never been as approachable [1, 2, 3, 4]. However, many of the canonical quantum algorithms developed over the last three decades remain unreachable in practice — requiring a large number of error corrected qubits and significant circuit depth. As a result, a new class of quantum algorithms — variational quantum algorithms (VQAs) [5, 6] — have come to shape the noisy intermediate-scale quantum (NISQ) era. First rising to prominence with the introduction of the variational quantum eigensolver (VQE) [7], they have evolved to cover topics such as optimization [8], quantum chemistry [9, 10, 11, 12, 13], integer factorization [14], compilation [15], quantum control [16], matrix diagonaliza-

tion [17, 18], and variational quantum machine learning [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31].

These algorithms have a common structure: a parametrized circuit is executed and a cost function is composed from expectation values measured in the resulting state. A classical optimization routine is then used to optimize the circuit parameters by minimizing said cost function. Initially, gradient-free optimization methods, such as Nelder-Mead and COBYLA, were common. However, gradient-based optimization provides significant advantages, from convergence guarantees [32] to the availability of workhorse algorithms (e.g., stochastic gradient descent) and software tooling developed for machine learning [33, 34, 35, 36, 37].

The so-called parameter-shift rule [16, 23, 38, 39] can be used to estimate the gradient for these optimization techniques, without additional hardware requirements and — in contrast to naïve numerical methods — without bias; the cost function is evaluated at two shifted parameter positions, and the rescaled difference of the results forms an unbiased estimate of the derivative. However, this two-term parameter-shift rule is restricted to gates with two distinct eigenvalues, potentially requiring expensive decompositions in order to compute hardware-compatible quantum gradients [40]. While various extensions to the shift rule have been discovered, they remain restricted to gates with a particular number of distinct eigenvalues [10, 41].

In this manuscript, we use the observation that the restriction of a variational cost function to a single parameter is a finite Fourier series [42, 43, 44, 45]; as a result, the restricted cost function can be *reconstructed* from circuit evaluations at shifted positions using a discrete Fourier transform (DFT). By analytically computing the derivatives of the Fourier series, we extract general parameter-shift rules for arbitrary quantum gates and provide closed-form expressions to apply them. In the specific case of unitaries with equidistant eigenvalues, the general parameter-shift rule recovers known parameter-shift rules from the literature, including the original two-term parameter-shift rule. We then generalize our approach in two steps: first from equidistant to arbitrary eigenvalues of the quantum gate, and from there — by making use of stochastic parameter shifts — to more complicated unitaries like multi-parameter gates. This enables us

David Wierichs: wierichs@thp.uni-koeln.de

Accepted in ⟨Quantum 2022-03-18, click title to verify. Published under CC-BY 4.0.
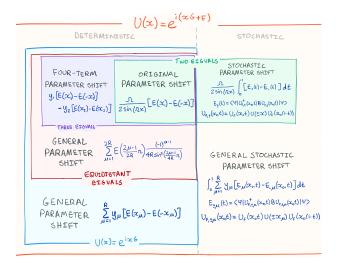
1

Figure 1: Overview of existing and new parameter-shift rules for first-order univariate derivatives as Venn diagram on the space of quantum gates. Each rule produces the analytic derivative for a set of gates, with more general rules reproducing the more specific ones. For gates of the form $U(x) = \exp(ixG)$ the rules are deterministic (*left*) whereas more involved gates of the form $U_F = \exp(i(xG + F))$ require stochastic evaluations of shifted values (*right*). See Sec. 2.2 for a summary of previously known shift rules. The fermionic four-term shift rule in Ref. [41] covers the same gates as the shown four-term rule (*purple*).

to cover *all* practically relevant quantum gates. An overview of the existing parameter-shift rules and our new results is shown in Fig. 1.

Afterwards, we perform an extensive resource analysis to compare the computational expenses required by both the general shift rule presented here, and decomposition-based approaches. In particular, we note that evaluating the cost of gradient recipes by comparing the number of unique executed circuits leads to fundamentally different conclusions on the optimal differentiation technique than when comparing the total number of measurements.

Our analysis not only is fruitful for understanding the structure of variational cost functions, but also has several practical advantages. Firstly, second-order derivatives (such as the Hessian [46] and the Fubini-Study metric tensor [47, 48]) can be computed with fewer evaluations compared to naïvely iterating the two-term parameter-shift rule. We also show, using the example of the *quantum approximate optimization algorithm* (QAOA), that the generalized parameter-shift rule can reduce the number of quantum circuit evaluations required for ansätze with repeated parameters.

Finally, we generalize the *quantum analytic descent* (QAD) algorithm [49] using the reconstruction of variational cost functions discussed here. We also reproduce the known generalizations of *Rotosolve* [50, 51] from single Pauli rotations to groups of rotations controlled by the same parameter [42, 45]; reconstruct-

ing functions with *arbitrary* spectrum extends this algorithm even further. Furthermore, the cost reduction for the gradient we present in the context of QAOA applies to Rotosolve as well. Similarly, future improvements that reduce the cost for gradient computations might improve the efficiency of these model-based algorithms, based on the analysis presented here.

This manuscript is structured as follows. In Sec. 2, we lay out the setting for our results by deriving the general functional form for variational cost functions, followed by a survey of existing parameter-shift rules. In Sec. 3 we show how to fully reconstruct univariate variational cost functions from a finite number of evaluations assuming an equidistant frequency spectrum, and derive parameter-shift rules for arbitrary-order univariate derivatives, including a generalization of the stochastic parameter-shift rule. In Sec. 4 we demonstrate how to compute second-order derivatives, in particular the Hessian and the metric tensor, more cheaply compared to existing methods. In Sec. 5 we discuss applications, applying the new generalized parameter-shift rules to QAOA, and using the full univariate reconstruction to extend existing model-based optimization methods. We end the main text in Sec. 6 with a discussion of our work and potential future directions. Finally, in the appendix we summarize some technical derivations (App. A), and extend the results to more general frequency spectra (App. B). The general stochastic parameter-shift rule and details on quantum analytic descent can be found in Apps. C and D.

Related work: In Ref. [42], the functions of VQAs were considered as Fourier series and parameter-shift rules were derived. Regarding the shift rules, the authors of Ref. [42] consider integer eigenvalues and derive a rule with $2R + 1$ evaluations for equidistant eigenvalues. In particular, the two-term and four-term shift rules are reviewed and formulated as special cases with *fewer* evaluations than the general result presented there. In contrast, our work results in the exact generalization of those shift rules, which requires $2R$ evaluations. Remarkably, Refs. [42, 45] also propose a generalized Rotosolve algorithm prior to its eponymous paper.

In addition, during the final stages of preparation of this work, a related work considering algebraic extensions of the parameter-shift rule appeared online [52]. The general description of quantum expectation values in Sec. 2.1 of the present work, along with its initial consequences in Sec. 3.1, are shown in Sec. II A of this preprint. We present a simpler derivation and further explore the implications this description has. The generalization of the parameter-shift rule in Ref. [52] is obtained by decomposing the gate generator using Cartan subalgebras, which can yield fewer shifted evaluations than decompositions of the gate itself. In particular, decompositions into

non-commuting terms, which do not lead to a gate decomposition into native quantum gates directly, can be used in this approach.

At a similar time, yet another work appeared [53], presenting a derivation similar to Sec. 2.1 and parameter-shift rules for the first order derivative. These rules are based on the ideas discussed here in Secs. 3.1 and 3.2.

2 Background

We start by deriving the form of a VQA cost function of a single parameter for a general single-parameter quantum gate. Then we review known parameter-shift rules and briefly discuss resource measures to compare these gradient recipes.

2.1 Cost functions arising from quantum gates

Let us first consider the expectation value for a general gate $U(x) = \exp(ixG)$, defined by a Hermitian generator G and parametrized by a single parameter x. Let $|\psi\rangle$ denote the quantum state that U is applied to, and B the measured observable[1]. The eigenvalues of $U(x)$ are given by $\{\exp(i\omega_j x)\}_{j\in[d]}$ with real-valued $\{\omega_j\}_{j\in[d]}$ where we denote $[d] := \{1, \ldots, d\}$ and have sorted the ω_j to be non-decreasing. Thus, we have:

$$E(x) := \langle\psi| U^\dagger(x) B U(x) |\psi\rangle \qquad (1)$$

$$= \sum_{j,k=1}^{d} \overline{\psi_j e^{i\omega_j x}} b_{jk} \psi_k e^{i\omega_k x} \qquad (2)$$

$$= \sum_{\substack{j,k=1 \\ j<k}}^{d} \left[\overline{\psi_j} b_{jk} \psi_k e^{i(\omega_k-\omega_j)x} \right. \qquad (3)$$
$$\left. + \psi_j \overline{b_{jk}\psi_k e^{i(\omega_k-\omega_j)x}} \right]$$
$$+ \sum_{j=1}^{d} |\psi_j|^2 b_{jj},$$

where we have expanded B and $|\psi\rangle$ in the eigenbasis of U, denoted by b_{jk} and ψ_j, respectively.

We can collect the x-independent part into coefficients $c_{jk} := \overline{\psi_j} b_{jk} \psi_k$ and introduce the R *unique positive* differences $\{\Omega_\ell\}_{\ell\in[R]} := \{\omega_k - \omega_j | j,k \in [d], \omega_k > \omega_j\}$. Note that the differences are not necessarily equidistant, and that for $r = |\{\omega_j\}_{j\in[d]}|$ *unique* eigenvalues of the gate generator, there are at most $R \leq \frac{r(r-1)}{2}$ unique differences. However, many quantum gates will yield $R \leq r$ *equidistant* differences in-

stead; a common example for this is

$$G = \sum_{k=1}^{\mathcal{P}} \pm P_k \qquad (4)$$

for commuting Pauli words P_k ($P_k P_{k'} = P_{k'} P_k$), which yields the frequencies $[\mathcal{P}]$ and thus $R = \mathcal{P}$.

In the following, we implicitly assume a mapping between the two indices $j, k \in [d]$ and the frequency index $\ell \in [R]$ such that $c_\ell = c_{\ell(j,k)}$ is well-defined[2]. We can then write the expectation value as a trigonometric polynomial (a finite-term Fourier series):

$$E(x) = a_0 + \sum_{\ell=1}^{R} c_\ell e^{i\Omega_\ell x} + \sum_{\ell=1}^{R} \overline{c_\ell} e^{-i\Omega_\ell x} \qquad (5)$$

$$= a_0 + \sum_{\ell=1}^{R} a_\ell \cos(\Omega_\ell x) + b_\ell \sin(\Omega_\ell x), \qquad (6)$$

with frequencies given by the differences $\{\Omega_\ell\}$, where we defined $c_\ell =: \frac{1}{2}(a_\ell - ib_\ell) \, \forall \ell \in [R]$ with $a_\ell, b_\ell \in \mathbb{R}$, and $a_0 := \sum_j |\psi_j|^2 b_{jj} \in \mathbb{R}$.

Since $E(x)$ is a finite-term Fourier series, the coefficients $\{a_\ell\}$ and $\{b_\ell\}$ can be obtained from a finite number of evaluations of $E(x)$ through a *discrete Fourier transform*. This observation (and variations thereof in Sec. 3) forms the core of this work: we can obtain the full functional form of $E(x)$ from a finite number of evaluations of $E(x)$, from which we can compute arbitrary order derivatives.

2.2 Known parameter-shift rules

Parameter-shift rules relate derivatives of a quantum function to evaluations of the function itself at different points. In this subsection, we survey known parameter-shift rules in the literature.

For functions of the form (6) with a single frequency $\Omega_1 = \Omega$ (i.e., G has two eigenvalues), the derivative can be computed via the parameter-shift rule [16, 23, 38]

$$E'(0) = \frac{\Omega}{2\sin(\Omega x_1)}[E(x_1) - E(-x_1)], \qquad (7)$$

where x_1 is a freely chosen shift angle from $(0, \pi)$ [3].

This rule was generalized to gates with eigenvalues $\{-1, 0, 1\}$, which leads to $R = 2$ frequencies, in Refs. [41, 10] in two distinct ways. The rule in Ref. [10] is an immediate generalization of the one above:

$$E'(0) = y_1[E(x_1) - E(-x_1)] \qquad (8)$$
$$- y_2[E(x_2) - E(-x_2)],$$

---

[1] Here we consider any pure state in the Hilbert space; in the context of VQAs, $|\psi\rangle$ is the state prepared by the subcircuit prior to $U(x)$. Similarly, B includes the subcircuit following up on $U(x)$.

[2] That is, $\ell(j,k) = \ell(j',k') \Leftrightarrow \omega_k - \omega_j = \omega_{k'} - \omega_{j'}$.

[3] The position 0 for the derivative is chosen for convenience but the rule can be applied at any position. To see this, note that shifting the argument of E does not change its functional form.

with freely chosen shift angles $x_{1,2}$ and corresponding coefficients $y_{1,2}$, requiring four evaluations to obtain $E'(0)$. A particularly symmetric choice of shift angles is $x_{1,2} = \pi/2 \mp \pi/4$ with coefficients $y_{1,2} = \frac{\sqrt{2}\pm 1}{2\sqrt{2}}$. In contrast, the rule in Ref. [41] makes use of an auxiliary gate to implement slightly altered circuits, leading to a structurally different rule:

$$E'(0) = \frac{1}{4}[E_+^+ - E_-^+ + E_+^- - E_-^-], \qquad (9)$$

where $E_\pm^\alpha$ is the measured energy when replacing the gate $U(x)$ in question by $U(x \pm \pi/2)\exp(\mp\alpha i\frac{\pi}{4}P_0)$ and P_0 is the projector onto the zero-eigenspace of the generator of U. Remarkably, this structure allows a reduction of the number of distinct circuit evaluations to two if the circuit and the Hamiltonian are real-valued, which is often the case for simulations of fermionic systems and forms a unique feature of this approach. This second rule is preferable whenever this condition is fulfilled, the auxiliary gates $\exp(\pm i\frac{\pi}{4}P_0)$ are available, and simultaneously the number of distinct circuits is the relevant resource measure.

Furthermore, the two-term parameter-shift rule Eq. (7) was generalized to gates with the more complicated gate structure $U_F(x) = \exp(i(xG + F))$ via the *stochastic parameter-shift rule* [39]

$$E'(x_0) = \frac{\Omega}{2\sin(\Omega x_1)}\int_0^1 [E_+(t) - E_-(t)]\mathrm{d}t. \quad (10)$$

Here, $E_\pm(t)$ is the energy measured in the state prepared by a modified circuit that splits $U_F(x_0)$ into $U_F(tx_0)$ and $U_F((1-t)x_0)$, and interleaves these two gates with $U_{F=0}(\pm x_1)$. See Sec. 3.6 and App. C for details. The first-order parameter-shift rules summarized here and their relationship to each other is also visualized in Fig. 1.

A parameter-shift rule for higher-order derivatives based on repeatedly applying the original rule has been proposed in Ref. [46]. The shift can be chosen smartly so that two function evaluations suffice to obtain the second-order derivative:

$$E''(0) = \frac{1}{2}[E(\pi) - E(0)], \qquad (11)$$

which like Eq. (7) is valid for single-frequency gates. Various expressions to compute combinations of derivatives with few evaluations were explored in Ref. [54].

2.3 Resource measures for shift rules

While the original parameter-shift rule Eq. (7) provides a unique, unbiased method to estimate the derivative $E'(0)$ via evaluations of E if it contains a single frequency, we will need to compare different shift rules for the general case. To this end, we consider two resource measures. Firstly, the number of distinct circuits that need to be evaluated to obtain all

terms of a shift rule, N_{eval}. This is a meaningful quantity on both, simulators that readily produce many measurement samples after executing each unique circuit once, as well as quantum hardware devices that are available via cloud services. In the latter case, quantum hardware devices are typically billed and queued per unique circuit, and as a result N_{eval} often dictates both the financial and time cost. Note that overhead due to circuit compilation and optimization scale with this quantity as well.

Secondly, we consider the overall number N of measurements — or *shots* — irrespective of the number of unique circuits they are distributed across. To this end, we approximate the physical (one-shot) variance σ^2 of the cost function E to be constant across its domain[4]. For an arbitrary quantity Δ computed from $\mathcal{M}$ values of E via a shift rule,

$$\Delta = \sum_\mu^{\mathcal{M}} y_\mu E(\boldsymbol{x}_\mu), \qquad (12)$$

we obtain the variance for the estimate of Δ as

$$\varepsilon^2 = \sum_\mu^{\mathcal{M}} |y_\mu|^2 \frac{\sigma^2}{N_\mu}, \qquad (13)$$

where N_μ expresses the number of shots used to measure $E(\boldsymbol{x}_\mu)$. For a total budget of N shots, the optimal shot allocation is $N_\mu = N|y_\mu|/\|\boldsymbol{y}\|_1$ such that

$$N = \frac{\sigma^2 \|\boldsymbol{y}\|_1^2}{\varepsilon^2}. \qquad (14)$$

This can be understood as the number of shots needed to compute Δ to a tolerable standard deviation ε.

The number of shots N is a meaningful quantity for simulators whose runtime scales primarily with the number of requested samples (e.g., Amazon Braket's TN1 tensor network simulator [1]), and for actual quantum devices when artificial resource measures like pricing per unique circuit and queueing time do not play a role.

In this work we will mostly use N_{eval} to compare the requirements of different parameter-shift rules as it is more accessible, does not rely on the assumption of constant physical variance like N does, and the coefficients $\boldsymbol{y}$ to estimate N are simply not known analytically in most general cases. For the case of equidistant frequencies and shift angles as discussed in Sec. 3.4 we will additionally compare the number of shots N in Sec. 3.5.

3 Univariate cost functions

In this section we study how a quantum cost function, which in general depends on multiple parameters, varies if only one of these parameters is changed.

---

[4]As it is impossible in general to compute σ^2 analytically, we are forced to make this potentially very rough approximation.

The results of this section will be sufficient to evaluate the gradient as well as the diagonal of the Hessian of a quantum function. We restrict ourselves to functions that can be written as the expectation value of an observable with respect to a state that is prepared using a unitary $U(x) = \exp(ixG)$ — capturing the full dependence on x. That is, all parameters but x are fixed and the operations they control are considered as part of the prepared state and the observable. As shown in Sec. 2.1, this yields a trigonometric polynomial, i.e.,

$$E(x) = a_0 + \sum_{\ell=1}^{R} a_\ell \cos(\Omega_\ell x) + b_\ell \sin(\Omega_\ell x). \quad (15)$$

In the following, we will assume the frequencies to be equidistant, i.e., $\Omega_\ell = \ell\Omega$, and generalize to arbitrary frequencies in App. B. While it is easy to construct gate sequences that do not lead to equidistant frequencies, many conventional gates and layers of gates do yield such a regular spectrum. The equidistant frequency case has two major advantages over the general case: we can derive closed-form parameter-shift rules (Sec. 3.4); and the number of circuits required for the parameter-shift rule scales much better (Sec. 3.5).

Without loss of generality, we further restrict the frequencies to integer values, i.e., $\Omega_\ell = \ell$. For $\Omega \neq 1$, we may rescale the function argument to achieve $\Omega_\ell = \ell$ and once we reconstruct the rescaled function, the original function is available, too.

3.1 Determining the full dependence on x

As we have seen, the functional form of $E(x)$ is known exactly. We can thus determine the function by computing the $2R+1$ coefficients $\{a_\ell\}$ and $\{b_\ell\}$. This is the well-studied problem of *trigonometric interpolation* (see e.g., [55, Chapter X]).

To determine $E(x)$ completely, we can simply evaluate it at $2R+1$ distinct points $x_\mu \in [-\pi, \pi)$. We obtain a set of $2R+1$ equations

$$E(x_\mu) = a_0 + \sum_{\ell=1}^{R} a_\ell \cos(\ell x_\mu) + b_\ell \sin(\ell x_\mu), \ \mu \in [2R]_0$$

where we denote $[2R]_0 := \{0, 1, \ldots, 2R\}$. We can then solve these linear equations for $\{a_\ell\}$ and $\{b_\ell\}$; this process is in fact a nonuniform *discrete Fourier transform (DFT)*.

A reasonable choice is $x_\mu = \frac{2\pi\mu}{2R+1}, \mu = -R, \ldots, R$, in which case the transform is the usual (uniform) DFT. For this choice, an explicit reconstruction for E follows directly from [55, Chapter X]; we reproduce it in App. A.1.1.

3.2 Determining the odd part of $E(x)$

It is often the case in applications that we only need to determine the odd part of E,

$$E_{\text{odd}}(x) = \frac{1}{2}(E(x) - E(-x)) \quad (16)$$

$$= \sum_{\ell=1}^{R} b_\ell \sin(\ell x). \quad (17)$$

For example, calculating odd-order derivatives of $E(x)$ at $x = 0$ only requires knowledge of $E_{\text{odd}}(x)$, since those derivatives of the even part vanish. Note that the reference point with respect to which E_{odd} is odd may be chosen arbitrarily, and does not have to be 0.

The coefficients in E_{odd} can be determined by evaluating E_{odd} at R distinct points x_μ with $0 < x_\mu < \pi$. This gives us a system of R equations

$$E_{\text{odd}}(x_\mu) = \sum_{\ell=1}^{R} b_\ell \sin(\ell x_\mu), \quad \mu \in [R] \quad (18)$$

which we can use to solve for the R coefficients $\{b_\ell\}$.

Using Eq. (16) we see that each evaluation of E_{odd} can be done with two evaluations of $E(x)$. Thus, the odd part of E can be completely determined with $2R$ evaluations of E, saving one evaluation compared to the general case. Note however that the saved $E(0)$ evaluation is evaluated regardless in many applications, and may be used to recover the full reconstruction — so, in effect, this saving does not have a significant impact[5].

3.3 Determining the even part of $E(x)$

We might similarly want to obtain the even part of E,

$$E_{\text{even}}(x) = \frac{1}{2}(E(x) + E(-x)) \quad (19)$$

$$= a_0 + \sum_{\ell=1}^{R} a_\ell \cos(\ell x), \quad (20)$$

which can be used to compute even-order derivatives of E.

Determining $E_{\text{even}}(x)$ requires $R+1$ evaluations of E_{even}, which leads to $2R+1$ evaluations of E for arbitrary frequencies. However, in the case where Ω_ℓ are integers, $R+1$ evaluations of E_{even} can be obtained

---

[5]If $E(0)$ is available, we can recover the full function, allowing us to, for example, evaluate its second derivative $E''(0)$ "for free". However, in practice many more repetitions may be needed for reasonable accuracy. This fact was already noted in [46] for the $R = 1$ case.

with $2R$ evaluations of $E(x)$ by using periodicity:

$$E_{\text{even}}(0) = E(0) \tag{21}$$

$$E_{\text{even}}(x_\mu) = \frac{1}{2}(E(x_\mu) + E(-x_\mu)), \tag{22}$$

$$0 < x_\mu < \pi, \ \mu \in [R-1]$$

$$E_{\text{even}}(\pi) = E(\pi). \tag{23}$$

Thus, in this case $2R$ evaluations of $E(x)$ suffice to determine its even part, saving one evaluation over the general case. In contrast to the odd part, this saving genuinely reduces the required computations as $E(0)$ is also used in the cheaper computation of $\{a_\ell\}$; therefore, if $E(0)$ is already known, we only require $2R - 1$ new evaluations.

We note that even though both the odd and the even part of $E(x)$ require $2R$ evaluations, the full function can be obtained at the price of $2R + 1$ evaluations.

3.4 Explicit parameter-shift formulas

Consider again the task of determining E_{odd} (E_{even}) based on its value at the shifted points $\{x_\mu\}$ with $\mu \in [R]$ ($\mu \in [R]_0$). This can be done by linearly combining elementary functions that vanish on all but one of the $\{x_\mu\}$, i.e., kernel functions, using the evaluation $E(x_\mu)$ as coefficients. If we restrict ourselves to evenly spaced points $x_\mu = \frac{2\mu-1}{2R}\pi$ ($x_\mu = \frac{\mu}{R}\pi$), we can choose these functions to be Dirichlet kernels. In addition to a straightforward reconstruction of the odd (even) function this delivers the *general parameter-shift rules*, which we derive in App. A.1:

$$E'(0) = \sum_{\mu=1}^{2R} E\left(\frac{2\mu-1}{2R}\pi\right) \frac{(-1)^{\mu-1}}{4R\sin^2\left(\frac{2\mu-1}{4R}\pi\right)}, \tag{24}$$

$$E''(0) = -E(0)\frac{2R^2+1}{6} + \sum_{\mu=1}^{2R-1} E\left(\frac{\mu\pi}{R}\right) \frac{(-1)^{\mu-1}}{2\sin^2\left(\frac{\mu\pi}{2R}\right)}. \tag{25}$$

We remark that derivatives of higher order can be obtained in an analogous manner, and with the same function evaluations for all odd (even) orders. Furthermore, this result reduces to the known two-term (Eq. (7)) and four-term (Eq. (8)) parameter-shift rules for $R = 1$ and $R = 2$, respectively, as well as the second-order derivative for $R = 1$ (Eq. (11)).

We again note that the formulas above use different evaluation points for the first and second derivatives ($2R$ evaluations for each derivative). Closed-form parameter-shift rules that use $2R + 1$ shared points can be obtained by differentiating the reconstruction formula Eq. (57).

3.5 Resource comparison

As any unitary may be compiled from (single-qubit) Pauli rotations, which satisfy the original parameter-shift rule, and CNOT gates, an alternative approach to compute $E'(0)$ is to decompose $U(x)$ into such gates and combine the derivatives based on the elementary gates. As rotation gates about any multi-qubit Pauli word satisfy the original parameter-shift rule as well, a more coarse-grained decomposition might be possible and yield fewer evaluations for this approach.

For instance, for the MAXCUT QAOA ansatz[6] on a graph $G = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V}$ and edges $\mathcal{E}$, one of the operations is to evolve under the problem Hamiltonian:

$$U_P(x) \propto \exp\left(-i\frac{x}{2}\sum_{(a,b)\in\mathcal{E}} Z_a Z_b\right) \tag{26}$$

$$= \prod_{(a,b)\in\mathcal{E}} \exp\left(-i\frac{x}{2}Z_a Z_b\right). \tag{27}$$

Eq. (26) treats $U_P(x)$ as a single operation with at most $M = |\mathcal{E}|$ frequencies $1, \ldots, R \leq M$, and we can apply the generalized parameter-shift rules of this section. Alternatively, we could decompose $U_P(x)$ with Eq. (27), apply the two-term parameter-shift rule to each R_{ZZ} rotation, and sum up the contributions using the chain rule.

3.5.1 Number of unique circuits

If there are $\mathcal{P}$ gates that depend on x in the decomposition, this approach requires $2\mathcal{P}$ unique circuit evaluations; as a result, the general parameter-shift rule is cheaper if $R < \mathcal{P}$. The evaluations used in the decomposition-based approach cannot be expressed by E directly because the parameter is shifted only in one of the $\mathcal{P}$ gates per evaluation, which makes the general parameter-shift rule more convenient and may reduce compilation overhead for quantum hardware, and the number of operations on simulators.

In order to compute $E''(0)$ via the decomposition, we need to obtain and sum the full Hessian of all elementary gates that depend on x (see App. A.4.2), which requires $2\mathcal{P}^2 - \mathcal{P} + 1$ evaluations, including $E(0)$, and thus is significantly more expensive than the $2R$ evaluations for the general parameter-shift rule.

While the derivatives can be calculated from the functional form of E_{odd} or E_{even}, the converse is not true for $R > 1$, i.e., the full functional dependence on x cannot be extracted from the first and second derivative alone. Therefore, the decomposition-based approach would demand a full multivariate reconstruction for all $\mathcal{P}$ parametrized elementary gates to obtain this dependence, requiring $\mathcal{O}(2^\mathcal{P})$ evaluations. The approach shown here allows us to compute the dependence in $2R + 1$ evaluations and thus is the only method for which the univariate reconstruction is viable.

---

[6]A more detailed description of the QAOA ansatz can be found in Sec. 5.1.

3.5.2 Number of shots

For equidistant evaluation points, we explicitly know the coefficients of the first and second-order shift rule given in Eqs. (24, 25), and thus can compare the variance of the derivatives in the context and under the assumptions of Sec. 2.3.

The coefficients satisfy (see App. A.4.1)

$$\sum_{\mu=1}^{2R} \left(4R \sin^2 \left(\frac{2\mu-1}{4R}\pi \right) \right)^{-1} = R$$

$$\frac{2R^2+1}{6} + \sum_{\mu=1}^{2R-1} \left(2\sin^2 \left(\frac{\mu\pi}{2R} \right) \right)^{-1} = R^2.$$

This means that the variance-minimizing shot allocation requires a shot budget of

$$N_{\text{genPS, 1}} = \frac{\sigma^2 R^2}{\varepsilon^2} \tag{28}$$

$$N_{\text{genPS, 2}} = \frac{\sigma^2 R^4}{\varepsilon^2} \tag{29}$$

using the generalized parameter-shift rule for the first and second derivative, respectively.

Assuming integer-valued frequencies in the cost function typically means, in the decomposition-based approach, that x enters the elementary gates without any additional prefactors[7]. Thus, optimally all evaluations for the first-order derivative rule are performed with the same portion of shots; whereas the second-order derivative requires an adapted shot allocation which, in particular, measures $E(0)$ with high precision as it enters $E''(0)$ with the prefactor $\mathcal{P}/2$. This yields (see App. A.4.2)

$$N_{\text{decomp, 1}} = \frac{\sigma^2 \mathcal{P}^2}{\varepsilon^2} \tag{30}$$

$$N_{\text{decomp, 2}} = \frac{\sigma^2 \mathcal{P}^4}{\varepsilon^2}. \tag{31}$$

Comparing with $N_{\text{genPS, 1}}$ and $N_{\text{genPS, 2}}$ above, we see that the shot budgets are equal at $\mathcal{P} = R$. That is, for both the first and second derivative, the general parameter-shift rule does not show lower shot requirements in general, in contrast to the previous analysis that showed a significantly smaller number of unique circuits for the second derivative. This shows that the comparison of recipes for gradients and higher-order derivatives crucially depends on the chosen resource measure. In specific cases we may be able to give tighter upper bounds on R so that $R < \mathcal{P}$ (see Sec. 5.1) and the general shift rule becomes favourable regarding the shot count as well.

3.6 General stochastic parameter-shift rule

Next, we will apply the *stochastic parameter-shift rule* to our general shift rule. For this section we do *not*

---

[7]Of course, one can construct less efficient decompositions that do not satisfy this rule of thumb.

---

assume the frequencies to be equidistant but address arbitrary spectra directly. Additionally we make the reference point x_0 at which the derivative is computed explicit.

In Ref. [39], the authors derive the stochastic parameter-shift rule for gates of the form

$$U_F(x) = \exp(i(xG + F)) \tag{32}$$

where G is a Hermitian operator with eigenvalues ± 1 (so that $G^2 = \mathbb{1}$), e.g., a Pauli word. F is any other Hermitian operator, which may not necessarily commute with G[8]. Key to the derivation of the stochastic rule is an identity relating the derivative of the quantum channel $\mathcal{U}_F(x)[\rho] = U_F^\dagger(x)\rho U_F(x)$ to the derivative of the generator channel $\mathcal{G}(x)[\rho] = i[(xG+F), \rho]$. We may extend this directly to the general parameter-shift rule for the case when $G^2 = \mathbb{1}$ is no longer satisfied (see App. C for the derivation):

$$E'(x_0) = \int_0^1 \sum_{\mu=1}^R y_\mu [E_\mu(x_0, t) - E_{-\mu}(x_0, t)] \mathrm{d}t \tag{33}$$

$$E_{\pm\mu}(x_0, t) := \langle B \rangle_{U_F(tx_0)U(\pm x_\mu)U_F((1-t)x_0)|\psi\rangle}.$$

The integration is implemented in practice by sampling values for t for each measurement of $E_\mu(x_0, t)$ and $E_{-\mu}(x_0, t)$.

The stochastic parameter-shift rule in combination with the generalized shift rule in Eq. (24) allows for the differentiation of any unitary with equidistant frequencies. As F in $U_F(x)$ above is allowed to contain terms that depend on other variational parameters, this includes multi-parameter gates in particular. Furthermore, combining Eq. (33) with the generalized shift rule for arbitrary frequencies in Eq. (90) allows us to compute the derivative of *any* quantum gate as long as the frequencies of $U_{F=0}(x)$ are known. We thus obtain an improved rule for $U_{F\neq 0}(x)$ over the original stochastic shift rule whenever the generalized shift rule is beneficial for $U(x) = U_{F=0}(x)$, compared to the decomposition-based approach.

4 Second-order derivatives

As noted in Sec. 3.3, higher-order derivatives of univariate functions are easily computed using the even or odd part of the function. In the following sections, we will extend our discussion to multivariate functions $E(\boldsymbol{x})$, where derivatives may be taken with respect to different variables. Each single parameter dependence is assumed to be of the form Eq. (5), with equidistant (and by rescaling integer-valued) frequencies $\{\Omega_\ell^{(k)}\}_{\ell \in [R_k]} = [R_k]$ for the kth parameter. We

---

[8]If $GF = FG$, the exponential may be split into $\exp(ixG)$ and $\exp(iF)$ and we are back at the situation $\exp(ixG)$.

---

may collect the numbers of frequencies in a vector $(\boldsymbol{R})_k = R_k$. It will again be useful in the following to make the reference point $\boldsymbol{x}_0$, at which these derivatives are computed, explicit.

4.1 Diagonal shift rule for the Hessian

Here we show how to compute the Hessian H of a multivariate function $E(\boldsymbol{x})$ at some reference point $\boldsymbol{x}_0$ using the Fourier series representation of E. We allow for single-parameter gates $U(x) = \exp(ixG)$ with equidistant frequencies and will use fewer evaluations of E than known schemes. An indication that this may be possible for gates with two eigenvalues was made in [54, Eq. (37)].

First, for the kth diagonal entry $H_{kk} = \partial_k^2 E(\boldsymbol{x}_0)$ of the Hessian, we previously noted in Sec. 3.3 that $2R_k$ evaluations are sufficient as it is the second derivative of a univariate restriction of E. Recall that one of the $2R_k$ evaluations is $E(\boldsymbol{x}_0)$; we can reuse this evaluation for all diagonal entries of H, and thus require $1 + \sum_{k=1}^{n}(2R_k - 1) = 2\|\boldsymbol{R}\|_1 - n + 1$ evaluations for the full diagonal. Further, if we compute the Hessian diagonal $(\boldsymbol{\nabla}^{\odot 2} E)_k := \partial_k^2 E$ in addition to the gradient, we may reuse the $2\|\boldsymbol{R}\|_1$ evaluations computed for the gradient, only requiring a single additional function value, namely $E(\boldsymbol{x}_0)$. In this case, we do not make use of the periodicity $E(\boldsymbol{x}_0 + \pi\boldsymbol{v}_k) = E(\boldsymbol{x}_0 - \pi\boldsymbol{v}_k)$, where $\boldsymbol{v}_k$ is the kth canonical basis vector, because this shift is not used in the gradient evaluation (see Sec. 3.2).

Next, for an off-diagonal entry $H_{km} = \partial_k\partial_m E(\boldsymbol{x}_0)$, consider the *univariate* trigonometric function that shifts the two parameters x_k and x_m *simultaneously*:

$$E^{(km)}(x) := E(\boldsymbol{x}_0 + x\boldsymbol{v}_{k,m}), \qquad (34)$$

where we abbreviated $\boldsymbol{v}_{k,m} := \boldsymbol{v}_k + \boldsymbol{v}_m$. We show in App. A.2 that $E^{(km)}$ again is a Fourier series of x with $R_{km} = R_k + R_m$ equidistant frequencies. This means that we can compute $E^{(km)\prime\prime}(0)$ via Eq. (25) with $R = R_{km}$, using $2R_{km} - 1$ evaluations of E (as we may reuse $E(\boldsymbol{x}_0)$ from the diagonal computation). Note that

$$\left.\frac{\mathrm{d}^2}{\mathrm{d}x^2}E^{(km)}(x)\right|_{x=0} = H_{kk} + H_{mm} + 2H_{km}, \qquad (35)$$

and that we have already computed the diagonal entries. We thus may obtain H_{km} via the *diagonal parameter-shift rule*

$$H_{km} = \frac{1}{2}\left(E^{(km)\prime\prime}(0) - H_{kk} - H_{mm}\right). \qquad (36)$$

In Fig. 2, we visually compare the computation of H_{km} via the diagonal shift rule to the chained application of univariate parameter-shift rules for x_k and x_m.

As an example, consider the case when $R_k = R_m = 1$ (e.g., where all parametrized gates are of the form
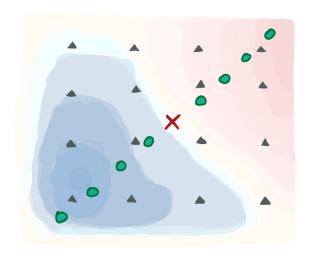

Figure 2: Visual representation of two approaches to compute a Hessian entry H_{km} at the position $\boldsymbol{x}_0$ (*red cross*). The parameters x_k and x_m lie on the coordinate axes and the heatmap displays the cost function $E(\boldsymbol{x})$. We may either combine the general shift rule for x_k and x_m (*grey triangles*) or compute the univariate derivative $E^{(km)\prime\prime}(0)$ and extract H_{km} via Eq. (36) (*green circles*).

$\exp(ix_k G_k/2)$ with $G_k^2 = \mathbb{1}$). By setting $R = 2$ in Eq. (25), we obtain the explicit formula for $E^{(km)\prime\prime}(0)$,

$$E^{(km)\prime\prime}(0) = -\frac{3}{2}E(\boldsymbol{x}_0) - \frac{1}{2}E(\boldsymbol{x}_0 + \pi\boldsymbol{v}_{k,m}) \qquad (37)$$
$$+ E\left(\boldsymbol{x}_0 + \frac{\pi}{2}\boldsymbol{v}_{k,m}\right) + E\left(\boldsymbol{x}_0 - \frac{\pi}{2}\boldsymbol{v}_{k,m}\right)$$

which can be combined with Eq. (36) to give an explicit formula for the Hessian. This formula (for $R_k = R_m = 1$) was already discovered in [54, Eq. (37)].

The computation of H_{km} along the main diagonal in the x_k-x_m-plane can be modified by making use of the second diagonal as well: define $\overline{\boldsymbol{v}}_{k,m} := \boldsymbol{v}_k - \boldsymbol{v}_m$ and $\overline{E}^{(km)}(x) := E(\boldsymbol{x}_0 + x\overline{\boldsymbol{v}}_{k,m})$, and compute

$$\left.\frac{\mathrm{d}^2}{\mathrm{d}x^2}\overline{E}^{(km)}(x)\right|_{x=0} = H_{kk} + H_{mm} - 2H_{km}, \qquad (38)$$
$$H_{km} = \frac{1}{4}\left(E^{(km)\prime\prime}(0) - \overline{E}^{(km)\prime\prime}(0)\right).$$

This means we can replace the dependence on the diagonal elements H_{kk} and H_{mm} by another univariate second-order derivative on the second diagonal. We will not analyze the resources required by this method in detail but note that for many applications it forms a compromise between the two approaches shown in Fig. 2.

We note that an idea similar to the ones presented here can be used for higher-order derivatives, but possibly requires more than one additional univariate reconstruction per derivative.

4.2 Resource comparison

For the Hessian computation, we will again look at the number of unique circuit evaluations N_{eval} and the number of shots N, as introduced in Sec. 2.3.

4.2.1 Number of unique circuits

In Tab. 1, we summarize the number of distinct circuit evaluations required to compute several combinations of derivatives of $E(\boldsymbol{x})$, either by decomposing the gate or by using the general parameter-shift rule together with the diagonal shift rule for the Hessian. We also include the generalized case of non-equidistant frequencies covered in App. B.2 for completeness. To obtain the cost for the repeated general shift rule, i.e., without the diagonal shift rule for the Hessian or decomposition, simply replace $\boldsymbol{\mathcal{P}}$ by $\boldsymbol{R}$ in the left column.

For equidistant frequencies, the diagonal shift rule for H_{km} requires $2(R_k + R_m) - 1$ evaluations, assuming the diagonal and thus $E(\boldsymbol{x}_0)$ to be known already. Like the gradient, H_{km} may instead be computed by decomposing $U_k(x_k)$ and $U_m(x_m)$ into $\mathcal{P}_k$ and $\mathcal{P}_m$ elementary gates, respectively, and repeating the parameter-shift rule twice [46, 56]. All combinations of parameter shifts are required, leading to $4\mathcal{P}_k\mathcal{P}_m$ evaluations. Finally, as a third option, one may repeat the general parameter-shift rule in Eq. (24) twice, leading to $4R_kR_m$ evaluations[9].

The repeated general shift rule requires strictly more circuit evaluations than the diagonal shift rule, since

$$2\|\boldsymbol{R}\|_1^2 - \|\boldsymbol{R}\|_1 + 1 > 2n\|\boldsymbol{R}\|_1 - \frac{1}{2}(n^2 + n - 2). \tag{39}$$

Similar to the discussion for the scaling of gradient computations, the optimal approach depends on $R_{k,m}$ and $\mathcal{P}_{k,m}$, but $\mathcal{P}$ and R often have a linear relation so that the diagonal shift rule will be significantly cheaper for many cost functions than decomposing the unitaries.

4.2.2 Number of shots

Next we compare the numbers of measurements required to reach a precision ε. While the approach via repeated shift rules uses distinct circuit evaluations for each Hessian entry, the diagonal shift rule in Eq. (36) reuses entries of the Hessian and thus correlates the optimal shot allocations and the statistical errors of the Hessian entries. We therefore consider an error measure on the full Hessian matrix instead of a single entry, namely the root mean square of the Frobenius norm of the difference between the true and the estimated Hessian. This norm is computed in

App. A.5 for the three presented approaches, and we conclude the number of shots required to achieve a norm of ε to be

$$N_{\text{diag}} = \frac{\sigma^2}{2\varepsilon^2}\left[\left(\sqrt{n+1} + n - 2\right)\|\boldsymbol{R}\|_2^2 + \|\boldsymbol{R}\|_1^2\right]^2 \tag{40}$$

$$N_{\text{genPS}} = \frac{\sigma^2}{2\varepsilon^2}\left[\left(\sqrt{2} - 1\right)\|\boldsymbol{R}\|_2^2 + \|\boldsymbol{R}\|_1^2\right]^2 \tag{41}$$

$$N_{\text{decomp}} = \frac{\sigma^2}{2\varepsilon^2}\left[\left(\sqrt{2} - 1\right)\|\boldsymbol{\mathcal{P}}\|_2^2 + \|\boldsymbol{\mathcal{P}}\|_1^2\right]^2 \tag{42}$$

In general, the diagonal shift rule for the Hessian is significantly less efficient than the repeated execution of the general parameter-shift rule if the shot count is the relevant resource measure. This is in sharp contrast to the number of unique circuits, which is strictly smaller for the diagonal shift rule. We note that the two resource measures yield *incompatible* recommendations for the computation of the Hessian. The overhead of the diagonal shift rule reduces to a (to leading order in n) constant prefactor if $R_k = R$ for all $k \in [n]$: in this case, we know $\|\boldsymbol{R}\|_1 = n = \|\boldsymbol{R}\|_2^2$ and therefore

$$\frac{N_{\text{diag}}}{N_{\text{genPS}}} = \frac{2n + \sqrt{n+1} - 2}{n + \sqrt{2} - 1} \xrightarrow{n\to\infty} 2. \tag{43}$$

4.3 Metric tensor

The Fubini-Study metric tensor $\mathcal{F}$ is the natural metric on the manifold of (parametrized) quantum states, and the key ingredient in quantum natural gradient descent [48]. The component of the metric belonging to the parameters x_k and x_m can be written as

$$\mathcal{F}_{km}(\boldsymbol{x}_0) = \Re\mathfrak{e}\{\langle\partial_k\psi(\boldsymbol{x})|\partial_m\psi(\boldsymbol{x})\rangle\}\Big|_{\boldsymbol{x}=\boldsymbol{x}_0} \tag{44}$$
$$- \langle\partial_k\psi(\boldsymbol{x})|\psi(\boldsymbol{x})\rangle\langle\psi(\boldsymbol{x})|\partial_m\psi(\boldsymbol{x})\rangle\Big|_{\boldsymbol{x}=\boldsymbol{x}_0},$$

or, alternatively, as a Hessian [46]:

$$\mathcal{F}_{km}(\boldsymbol{x}_0) = -\frac{1}{2}\partial_k\partial_m|\langle\psi(\boldsymbol{x})|\psi(\boldsymbol{x}_0)\rangle|^2\Big|_{\boldsymbol{x}=\boldsymbol{x}_0}$$
$$=: \partial_k\partial_m f(\boldsymbol{x}_0). \tag{45}$$

It follows that we can compute the metric using the same method as for the Hessian, with $f(\boldsymbol{x})$ as the cost function. We know the value of f without shift as

$$f(\boldsymbol{x}_0) = -\frac{1}{2}|\langle\psi(\boldsymbol{x}_0)|\psi(\boldsymbol{x}_0)\rangle|^2 = -\frac{1}{2}. \tag{46}$$

The values with shifted argument can be calculated as the probability of the zero bitstring $\boldsymbol{0}$ when measuring the state $V^\dagger(\boldsymbol{x})V(\boldsymbol{x}_0)|\boldsymbol{0}\rangle$ in the computational basis, which requires circuits with up to doubled depth compared to the original circuit $V(\boldsymbol{x})$. Alternatively, we may use a Hadamard test to implement f, requiring an auxiliary qubit, two operations controlled by that qubit as well as a measurement on it, but only

---

[9]These $4R_kR_m$ shifted evaluations are *not* simultaneous shifts in both directions of the form Eq. (34).

Accepted in 〈 〉uantum 2022-03-18, click title to verify. Published under CC-BY 4.0.

9

Quantity	Decomposition	Gen. shift rule, equidistant	Gen. shift rule
$E(\boldsymbol{x}_0)$	1	1	1
$\partial_k E(\boldsymbol{x}_0)$	$2\mathcal{P}_k$	$2R_k$	$2R_k$
$\boldsymbol{\nabla} E(\boldsymbol{x}_0)$	$2\|\boldsymbol{\mathcal{P}}\|_1$	$2\|\boldsymbol{R}\|_1$	$2\|\boldsymbol{R}\|_1$
$\partial_k^2 E(\boldsymbol{x}_0)$	$2\mathcal{P}_k^2 - \mathcal{P}_k + 1$	$2R_k$	$2R_k + 1$
$\boldsymbol{\nabla}^{\odot 2} E(\boldsymbol{x}_0)$	$2\|\boldsymbol{\mathcal{P}}\|_2^2 - \|\boldsymbol{\mathcal{P}}\|_1 + 1$	$2\|\boldsymbol{R}\|_1 - n + 1$	$2\|\boldsymbol{R}\|_1 + 1$
$\partial_k \partial_m E(\boldsymbol{x}_0)$	$4\mathcal{P}_k\mathcal{P}_m$	$2(R_k + R_m) - 1^{(*)}$	$4R_k R_m + 2R_k + 2R_m - 4^{(*)}$
$\boldsymbol{\nabla}^{\otimes 2} E(\boldsymbol{x}_0)$	$2\|\boldsymbol{\mathcal{P}}\|_1^2 - \|\boldsymbol{\mathcal{P}}\|_1 + 1$	$2n\|\boldsymbol{R}\|_1 - \frac{1}{2}(n^2 + n - 2)$	$\begin{aligned} &2\left(\|\boldsymbol{R}\|_1^2 - \|\boldsymbol{R}\|_2^2 + n\|\boldsymbol{R}\|_1\right)\\ &\qquad -2n(n-1)+1 \end{aligned}$
$\partial_k E(\boldsymbol{x}_0)$ & $\partial_k^2 E(\boldsymbol{x}_0)$	$2\mathcal{P}_k^2 + 1$	$2R_k + 1$	$2R_k + 1$
$\boldsymbol{\nabla} E(\boldsymbol{x}_0)$ & $\boldsymbol{\nabla}^{\odot 2} E(\boldsymbol{x}_0)$	$2\|\boldsymbol{\mathcal{P}}\|_2^2 + 1$	$2\|\boldsymbol{R}\|_1 + 1$	$2\|\boldsymbol{R}\|_1 + 1$
$\boldsymbol{\nabla} E(\boldsymbol{x}_0)$ & $\boldsymbol{\nabla}^{\otimes 2} E(\boldsymbol{x}_0)$	$2\|\boldsymbol{\mathcal{P}}\|_1^2 + 1$	$2n\|\boldsymbol{R}\|_1 - \frac{1}{2}(n^2 - n - 2)$	$\begin{aligned} &2\left(\|\boldsymbol{R}\|_1^2 - \|\boldsymbol{R}\|_2^2 + n\|\boldsymbol{R}\|_1\right)\\ &\qquad -2n(n-1)+1 \end{aligned}$

Table 1: Number of distinct circuit evaluations N_{eval} for measuring combinations of derivatives of a parametrized expectation value function E at parameter position $\boldsymbol{x}_0$. The compared approaches include decomposition of the unitaries together with the original parameter-shift rule (*left*), and the generalized parameter-shift rule Eq. (24) together with the diagonal shift rule for the Hessian in Eq. (36). The requirements for the latter differ significantly for equidistant (*center*) and arbitrary frequencies (*right*, see App. B.2). A third approach is to repeat the general parameter-shift rule, the cost of which can be read off by replacing $\boldsymbol{\mathcal{P}}$ by $\boldsymbol{R}$ in the left column. Here, n is the number of parameters in the circuit, $\mathcal{P}_k$ is the number of elementary gates with two eigenvalues in the decomposition of the kth parametrized unitary, and R_k denotes the number of frequencies for the kth parameter. The asterisk $^{(*)}$ indicates that the derivatives $\partial_k^2 E$ and $\partial_m^2 E$ need to be known in order to obtain the mixed derivative at the shown price (see main text). The evaluation numbers take savings into account that are based on using evaluated energies for multiple derivative quantities; hence, they are not additive in general.

halved depth on average (see App. A.3). With either of these methods, the terms for the shift rule in Eq. (36) and thus the metric tensor can be computed via the parameter-shift rule.

The metric can also be computed analytically without parameter shifts via a *linear combination of unitaries (LCU)* [57, 58], which also employs Hadamard tests. As it uses the generator as an operation in the circuit, any non-unitary generator needs to be decomposed into Pauli words for this method to be available on quantum hardware, similar to a gate decomposition. Afterwards, this method uses one circuit evaluation per pair of Pauli words from the kth and mth generator to compute the entry $\mathcal{F}_{km}$. A modification of all approaches that use a Hadamard test is possible by replacing it with projective measurements [56].

Metric entries that belong to operations that commute *within the circuit*[10] can be computed block-wise without any auxiliary qubits, additional operations or deeper circuits [48]. For a given block, we execute the subcircuit V_1 prior to the group of mutually commuting gates and measure the covariance matrix of the generators $\{G_k\}$ of these gates:

$$\mathcal{F}_{km} = \langle \boldsymbol{0}| V_1^\dagger G_k G_m V_1 |\boldsymbol{0}\rangle \tag{47}$$
$$- \langle \boldsymbol{0}| V_1^\dagger G_k V_1 |\boldsymbol{0}\rangle \langle \boldsymbol{0}| V_1^\dagger G_m V_1 |\boldsymbol{0}\rangle .$$

By grouping the measurement bases of all $\{G_k G_m\}$

and $\{G_k\}$ of the block, the covariance matrix can typically be measured with only a few unique circuit evaluations[11], making this method the best choice for the block-diagonal. One may then either use the result as an approximation to the full metric tensor, or use one of the other methods to compute the off-block-diagonal entries; the approximation has been shown to work well for some circuit structures [48], but not for others [59]. The methods to obtain the metric tensor and their resource requirements are shown in Tab. 2.

Since we run a different circuit for the metric tensor than for the cost function itself, the $2R_k - 1$ evaluations at shifted positions needed for the kth diagonal entry cannot reuse any prior circuit evaluations, as is the case for the cost function Hessian. Consequentially, the natural gradient of a (single term) expectation value function E,

$$\boldsymbol{\nabla}_{\text{n}} E(\boldsymbol{x}) := \mathcal{F}^{-1}(\boldsymbol{x})\boldsymbol{\nabla}E(\boldsymbol{x}), \tag{48}$$

with $\boldsymbol{\nabla}E$ referring to the Euclidean gradient, requires more circuit evaluations than its Hessian and gradient together.

However, the utility of the metric tensor becomes apparent upon observing that it depends solely on the *ansatz*, and not the observable being measured. This

---

[10]For example, operations on distinct wires commute in general but not necessarily within the circuit if entangling operations are carried out between them.

[11]For a layer of simultaneous single-qubit rotations on all N qubits, even a single measurement basis is sufficient for the corresponding $N \times N$ block.

	Parameter shift rule		LCU	Covariance
	Overlap	Hadamard		
Aux. qubits	0	1	1	0
off-block-diag.	✓	✓	✓	
Depth (avg)	$\sim \frac{4}{3}D_V$	$\sim \frac{2}{3}D_V$	$\sim \frac{2}{3}D_V$	$\frac{2}{3}D_V$
Depth (max)	$2D_V$	$\sim D_V$	$\sim D_V$	D_V
$N_{\text{eval}}(\mathcal{F}_{kk})$	$\begin{cases} 2R_k - 1 \\ 2R_k \end{cases}$		$\mathcal{Q}_k \le \frac{1}{2}(\mathcal{P}_k^2 - \mathcal{P}_k)$	$\overline{\mathcal{P}}_k \le \mathcal{P}_k$
$N_{\text{eval}}(\mathcal{F}_{km})$	$\begin{cases} 2(R_k + R_m) - 1 \\ 2(2R_k R_m + R_k + R_m - 2) \end{cases}$		$\mathcal{P}_k \mathcal{P}_m$	$\overline{\mathcal{P}}_{km} \le \mathcal{P}_k \mathcal{P}_m$
$N_{\text{eval}}(\mathcal{F})$	$\begin{cases} 2n\|\boldsymbol{R}\|_1 - \frac{1}{2}(n^2 + n) \\ 2\left(\|\boldsymbol{R}\|_1^2 - \|\boldsymbol{R}\|_2^2 + n(\|\boldsymbol{R}\|_1 - n + 1)\right) \end{cases}$		$\frac{1}{2}\left(\|\boldsymbol{\mathcal{P}}\|_1^2 - \|\boldsymbol{\mathcal{P}}\|_2^2\right) + \|\boldsymbol{\mathcal{Q}}\|_1$	—

Table 2: Quantum hardware-ready methods to compute the Fubini-Study metric tensor and their resource requirements. The cost function $f(\boldsymbol{x})$ (see Eq. (45)) for the parameter-shift rule can be implemented with increased depth by applying the adjoint of the original circuit to directly realize the overlap (*left*) or with an auxiliary qubit and Hadamard tests (*center left*, App. A.3). The LCU method (*center right*) is based on Hadamard tests as well and both these methods can spare the auxiliary qubit and instead employ projective measurements [56]. The cheapest method is via measurements of the covariance of generators (*right*) but it can only be used for the block-diagonal of the tensor, i.e., not for all $\mathcal{F}_{km}$. We denote the depth of the original circuit V by D_V and the number of Pauli words in the decomposition of G_k and its square with $\mathcal{P}_k$ and $\mathcal{Q}_k$, respectively. The $\mathcal{P}_k$ Pauli words of G_k can be grouped into $\overline{\mathcal{P}}_k$ groups of pairwise commuting words; the number of groups of pairwise commuting Pauli words in the product $G_k G_m$ similarly is $\overline{\mathcal{P}}_{km}$. For the covariance-based approach, we overestimate the number of required circuits, as typically many of the measurement bases of the entries in the same block will be compatible. The number of unique circuits to be evaluated for a diagonal element $\mathcal{F}_{kk}$, an off-diagonal element $\mathcal{F}_{km}$, and the full tensor $\mathcal{F}$ is given in terms of the number of frequencies R_k and of $\mathcal{Q}_k$, $\mathcal{P}_k$ $\overline{\mathcal{P}}_k$ and $\overline{\mathcal{P}}_{km}$. The entries for N_{eval} in the first and second row of the braces refer to equidistant (main text) and arbitrary frequencies (see App. B.2), respectively.

means that if a cost function has multiple terms, like in VQEs, the metric only needs to be computed once per epoch, rather than once per term, as is the case of the cost function Hessian. Therefore, an epoch of quantum natural gradient descent can be cheaper for such cost functions than an epoch of optimizers using the Hessian of the cost function. In addition, the block-diagonal of the metric tensor can be obtained with few circuit evaluations per block for conventional gates without any further requirements and with reduced average circuit depth.

5 Applications

In this section, we will present QAOA as concrete application for our general parameter-shift rule, which reduces the required resources significantly when computing derivatives. Afterwards, we use the approach of trigonometric interpolation to generalize the Rotosolve algorithm. This makes it applicable to arbitrary quantum gates with equidistant frequencies, which reproduces the results in Refs. [42, 45], and extends them further to more general frequency spectra. In addition, we make quantum analytic descent (QAD) available for arbitrary quantum gates with equidistant frequencies, which previously required a higher-dimensional Fourier reconstruction and thus was infeasible.

5.1 QAOA and Hamiltonian time evolution

In Eq. (24) we presented a generalized parameter-shift rule that makes use of $2R$ function evaluations for R frequencies in E. A particular example for single-parameter unitaries with many frequencies are layers of single- or two-qubit rotation gates, as can be found e.g., in QAOA circuits or digitized Hamiltonian time evolution algorithms.

The quantum approximate optimization algorithm (QAOA) was first proposed in 2014 by Farhi, Goldstone and Gutmann to solve classical combinatorial optimization problems on near-term quantum devices [8]. Since then, it has been investigated analytically [60, 61, 62], numerically [63, 64], and on quantum computers [65, 66].

In general, given a problem Hamiltonian H_P that encodes the solution to the problem of interest onto N qubits, QAOA applies two types of layers alternatingly to an initial state $|+\rangle^{\otimes N}$:

$$V_{\text{QAOA}}(\boldsymbol{x}) = \prod_{j=p}^{1} U_M(x_{2j}) U_P(x_{2j-1}), \quad (49)$$

where p is the number of blocks which determines the depth of the circuit, $U_M(x) = \exp\left(-ixH_M\right)$ with $H_M = \sum_{k=1}^{N} X_k$ is the so-called *mixing layer*, and $U_P(x) = \exp(-ixH_P)$ is the time evolution under H_P. The parameters $\boldsymbol{x}$ can then be optimized to try to

minimize the objective function

$$E(\boldsymbol{x}) = \langle+|^{\otimes N} V_{\text{QAOA}}^{\dagger}(\boldsymbol{x}) H_P V_{\text{QAOA}}(\boldsymbol{x}) |+\rangle^{\otimes N}. \quad (50)$$

Here we focus on the layer U_P, and we look at the example of MAXCUT in particular. The corresponding problem Hamiltonian for an unweighted graph $G = (\mathcal{V}, \mathcal{E})$ with N vertices $\mathcal{V}$ and M edges $\mathcal{E}$ reads

$$H_P = \sum_{(a,b)\in\mathcal{E}} \frac{1}{2}(1 - Z_a Z_b), \quad (51)$$

and U_P correspondingly contains M two-qubit Pauli-Z rotations R_{ZZ}.

We note that H_M has eigenvalues $-N, -N + 2, \cdots, N$, which means the corresponding frequencies (differences of eigenvalues) are $2, \cdots, 2N$. Thus, treating $U_M(x_{2j})$ as a single operation, Eq. (6) implies that $E(\boldsymbol{x})$ can be considered an N-order trigonometric polynomial in x_{2j}, and the parameter-shift rules we derive in Sec. 3 will apply with $R = N$. Similarly, H_P has corresponding frequencies in the set $[M]$, and it will obey the parameter-shift rule for $R = M$, although we may be able to give better upper bounds λ for R. Thus the unique positive differences $\{\Omega_\ell\}$ for those layers, i.e., the frequencies of $E(\boldsymbol{x})$ with respect to the parameter $\{x_{2j-1}\}_{j\in[p]}$, take integer values within the interval $[0, \lambda]$ as well. We may therefore use Eq. (24), with the knowledge that $R \leq \lambda \leq M$.

Note that knowing *all* frequencies of $E(x)$ requires knowledge of the full spectrum of H_P — and in particular of λ — which in turn is the solution of MAXCUT itself. As a consequence, the motivation for performing QAOA becomes obsolete. Therefore, in general we cannot assume to know $\{\Omega_\ell\}$ (or even R), but instead require upper bounds $\varphi(G) \geq \text{MAXCUT}(G) = \lambda$ which can be used to bound the largest frequency, and thus the number of frequencies R and subsequently the number of terms in the parameter-shift rule. It is noteworthy that even if the *largest* frequency λ is known exactly via a tight bound — which restricts the Fourier spectrum to the integers $[\lambda]$ — not *all* integers smaller than λ need to be present in the set of frequencies $\{\Omega_\ell\}$, so that the estimate for R may be too large[12].

One way to obtain an upper bound uses analytic results based on the Laplacian of the graph of interest [67, 68], for which automatic bound generating programs exist [69]. An alternative approach uses semi-definite programs (SDPs) that solve relaxations of the MAXCUT problem, the most prominent being the *Goemans-Williamson (GW)* algorithm [70] and recent extensions thereof that provide tighter upper bounds [71, 72]. The largest eigenvalue is guaranteed to be within ~ 0.878 of these SDP upper bounds.

To demonstrate the above strategy, we summarize the number of evaluations required for the gradient and Hessian of an n-parameter QAOA circuit on N qubits for MAXCUT in Tab. 3, comparing the approach via decomposing the circuit, to the one detailed above based on φ and the improved Hessian measurement scheme in Sec. 4.1. Here, we take into account that half of the layers are of the form U_P, and the other half are mixing layers with $R = N$ frequencies. We systematically observe the number of evaluations for the gradient to be cut in half, and the those for the gradient and Hessian together to scale with halved order in N (and k, for regular graphs).

In addition, we display the numbers of circuit evaluations from Tab. 3 together with SDP-based bounds for λ and the true minimal number of evaluations required for the parameter-shift rule in Fig. 3. For this, we sampled random unweighted graphs of the corresponding type and size and ran the GW algorithm as well as an improvement thereof to obtain tighter bounds [71]. On one hand we observe the advantage of the generalized parameter-shift rule and the cheaper Hessian method that can be read off already from the scalings in Tab. 3. On the other hand, we find both SDP-based upper bounds to provide an exact estimate of the largest eigenvalue in the $N \leq 20$ regime, as can be seen from the cut values obtained from the GW algorithm that coincide with the upper bound. In cases in which the frequencies $\{\Omega_\ell\}$ occupy all integers in $[R]$, this leads to an exact estimate of R and the evaluations in the shift rule. For all graph types but complete graphs, the SDP-based upper bounds yield a better estimate for the number of terms than the respective analytic bound φ, which improves the generalized shift rule further.

In summary, we find the generalized parameter-shift rule to offer a constant prefactor improvement when computing the gradient and an improvement of at least $\mathcal{O}(N)$ when computing both the gradient and the Hessian. For certain graph types, knowledge about the structure of the spectrum and tight analytic bounds provide this advantage already, whereas for other graph types the SDP-based bounds reduce the evaluation numbers significantly.

5.2 Rotosolve

The *Rotosolve* algorithm is a coordinate descent algorithm for minimizing quantum cost functions. It has been independently discovered multiple times [42, 45, 51, 50], with [50] giving the algorithm its name but only (along with [51]) considering parametrized Pauli rotations, and [42, 45] covering other unitaries with integer-valued generator eigenvalues.

The Rotosolve algorithm optimizes the rotation angles sequentially: for one variational parameter x_k at a time, the cost function is reconstructed as a function of that parameter using $2R_k + 1$ evaluations, the mini-

---

[12]A simple example for this is the case of $2k$-regular graphs; here, H_P only has even eigenvalues, and therefore all frequencies are even as well. Given an upper bound φ, we thus know the number of frequencies to satisfy $R \leq \varphi/2$.

Graph type	Decomposition-based		Gen. shift rule		
	$\boldsymbol{\nabla} E$	$\boldsymbol{\nabla} E \& \boldsymbol{\nabla}^{\otimes 2} E$	Bound φ	$\boldsymbol{\nabla} E$	$\boldsymbol{\nabla} E \& \boldsymbol{\nabla}^{\otimes 2} E$
General	$(M+N)n$	$\mathcal{O}(n^2(M+N)^2)$	φ	$n(\varphi+N)$	$\mathcal{O}(n^2(\varphi+N))$
Complete	$\frac{1}{2}n(N^2+N)$	$\mathcal{O}(n^2N^4)$	$\left\lfloor \frac{N^2}{4} \right\rfloor$	$n\left(\left\lfloor \frac{N^2}{4} \right\rfloor + N \right)$	$\mathcal{O}(n^2N^2)$
$2k$-regular	$(k+1)nN$	$\mathcal{O}(k^2n^2N^2)$	kN	$\frac{k+2}{2}nN$	$\mathcal{O}(kn^2N)$
$(2k+1)$-regular	$\frac{2k+3}{2}nN$	$\mathcal{O}(k^2n^2N^2)$	$\frac{2k+1}{2}N$	$\frac{2k+3}{2}nN$	$\mathcal{O}(kn^2N)$

Table 3: Evaluation numbers for the gradient, or both the gradient and the Hessian, for QAOA circuits for MAXCUT on several types of graphs. Each graph has N vertices and a graph type-specific number M of edges, and the (even) number of parameters is denoted as n. For K-regular graphs, we know $M = \min\{(N^2 - N)/2, KN/2\}$, and the latter value is used in the displayed evaluation costs; if the former value forms the minimum, the graph is in fact complete. The left column is based on decomposing the circuit, applying the conventional two-term parameter-shift rule per elementary gate and iterating it for the Hessian. The right column employs the generalized parameter-shift rule Eq. (24) combined with an upper bound φ for the largest eigenvalue λ of the problem Hamiltonian, as well as the reduced number of evaluations for Hessian off-diagonal terms from Sec. 4.1. The bound φ for complete graphs can be found in Ref. [67].

mum of the reconstruction is calculated, and then the parameter is updated to the minimizing angle. For the case of Pauli rotation gates this minimum can be found via a closed-form expression. Recent studies have shown such coordinate descent methods to work well on many tasks [73, 50, 45, 74], although there are limited cases where these methods fail [75].

While Rotosolve is not gradient-based, our cost reduction for the gradient presented in Sec. 5.1 stems from a cost reduction for function reconstruction, and hence is applicable to Rotosolve as well.

As shown in Sec. 3.1, the univariate objective function can also be fully reconstructed if the parametrized unitaries are more complicated than Pauli rotations, using the function value itself and the evaluations from the generalized parameter-shift rule. Since the generalized parameter-shift rule also applies for non-equidistant frequencies (see App. B), the reconstruction works in the same way for arbitrary single-parameter gates. This extends our generalization of Rotosolve beyond the previously known integer-frequency case [42, 45], although the number of frequencies—and thus the cost—for the reconstruction are typically significantly increased for non-integer frequencies. While the minimizing angle might not be straightforward to express in a closed form as it is the case for a single frequency, the one-dimensional minimization can efficiently be carried out numerically to high precision, via grid search or semi-definite programming [76, Chapter 4.2].

5.3 Quantum analytic descent

Quantum analytic descent (QAD) [49] also approaches the optimization problem in VQAs via trigonometric interpolation. In contrast to Rotosolve, it considers a model of all parameters *simultaneously* and includes second-order derivatives, but this model only is a *local approximation* of the full cost function. Additionally, QAD has been developed for circuits that exclusively contain Pauli rotations as

parametrized gates.

The algorithm evaluates the cost function E at $2n^2 + n + 1$ points around a reference point $\boldsymbol{x}_0$, and then constructs a trigonometric model of the form[13]

$$\hat{E}(\boldsymbol{x}_0 + \boldsymbol{x}) = A(\boldsymbol{x}) \left[E^{(A)} + 2\boldsymbol{E}^{(B)} \cdot \tan\left(\frac{\boldsymbol{x}}{2}\right) \right.$$
$$+ 2\boldsymbol{E}^{(C)} \cdot \tan\left(\frac{\boldsymbol{x}}{2}\right)^{\odot 2} \qquad (52)$$
$$\left. + 4\tan\left(\frac{\boldsymbol{x}}{2}\right) \cdot E^{(D)} \cdot \tan\left(\frac{\boldsymbol{x}}{2}\right) \right],$$

Here, we introduced $A(\boldsymbol{x}) := \prod_k \cos^2\left(\frac{x_k}{2}\right)$ and the element-wise square of a vector $\boldsymbol{v}$, $(\boldsymbol{v}^{\odot 2})_k := v_k^2$ as for the Hessian diagonal. The coefficients $E^{(A/B/C/D)}$ are derived from the circuit evaluations, taking the form of a scalar, two vectors and an upper triangular matrix. More precisely, the expansion basis is chosen such that $\boldsymbol{E}^{(B)} = \boldsymbol{\nabla}E(\boldsymbol{x}_0)$, $\boldsymbol{E}^{(C)} = \boldsymbol{\nabla}^{\odot 2}E(\boldsymbol{x}_0)$, and $E^{(D)}$ is the strictly upper triangular part of the Hessian. Note that for this model $2n^2 + n + 1$ evaluations are used to obtain $n^2/2 + 3n/2 + 1$ parameters. In the presence of statistical noise from these evaluations, it turns out that building the model to a desired precision and inferring modelled gradients close to the reference point $\boldsymbol{x}_0$ has resource requirements similar to measuring the gradient directly [49].

This model coincides with $E(\boldsymbol{x})$ at $\boldsymbol{x}_0$ up to second order, and in the vicinity its error scales with the third order of the largest parameter deviation [49]. After the construction phase, the model cost is minimized in an inner optimization loop, which only requires classical operations. For an implementation and demonstration of the optimization, we also refer the reader to [77] and [78].

In the light of the parameter-shift rules and reconstruction methods, we propose three (alternative) modifications of QAD. The first change is to reduce the required number of evaluations. As the coeffi-

---
[13]We slightly modify the trigonometric basis functions from Ref. [49] to have leading order coefficients 1.

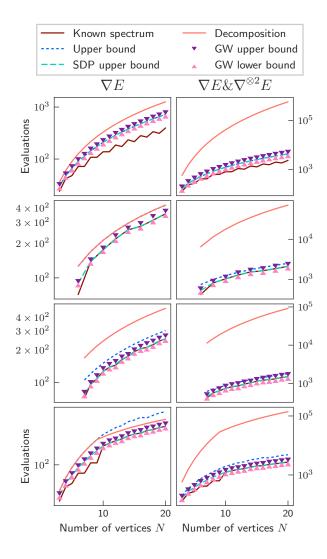Figure 3: Evaluation numbers N_{eval} for the gradient (*left*) or both the gradient and the Hessian (*right*) for $n = 6$ parameter QAOA circuits for MAXCUT on graphs of several types and sizes. Using numerical upper bounds together with our new parameter-shift rule (GW – *purple triangles* and its generalization – *dashed turquoise*) reduces the resource requirements for both quantities significantly, compared to the previously available decomposition-based method (*solid orange*). The rows correspond to the various considered graph types (*top to bottom*): complete, 5-regular, 6-regular and (up to) $4N$ randomly sampled edges. The requirements for the decomposition-based approach and the analytic upper bound (*dotted blue*) correspond to the results in the left and right column of Tab. 3, respectively. The numerical *upper* bounds both use the minimized objective value of SDPs for relaxations of MAXCUT to obtain the bound φ, which depends on the graph instance. The GW-based *lower* bound (*pink triangles*) is obtained by randomly mapping the output state of the GW algorithm to 10 valid cuts and choosing the one with the largest cut value. Note that K-regular graphs are only defined for $N > K$ and $NK \mod 2 = 0$ and that graphs with κN sampled edges are complete for $N \leq 2\kappa + 1$, leading to a change in the qualitative behaviour in the last row at $N = 2\kappa + 2 = 10$.

cients $E^{(A/B/C/D)}$ consist of the gradient and Hessian, they allow us to exploit the reduced resource requirements presented in Tab. 1 [14]. In the case originally considered by the authors, i.e., for Pauli rotations only, this reduces the number of evaluations from $2n^2 + n + 1$ to $(3n^2 + n)/2 + 1$.

A second, alternative modification of QAD is to keep all evaluations as originally proposed to obtain the full second-order terms, i.e., we may combine the shift angles for each pair of parameters, and use them for coefficients of additional higher-order terms. This extended model (see App. D.1) has the form

$$\mathring{E}(\boldsymbol{x}_0 + \boldsymbol{x}) = \hat{E}(\boldsymbol{x}_0 + \boldsymbol{x}) + 4A(\boldsymbol{x})\tan\left(\frac{\boldsymbol{x}}{2}\right)^{\odot 2} \quad (53)$$
$$\cdot \left[E^{(F)} \cdot \tan\left(\frac{\boldsymbol{x}}{2}\right) + E^{(G)} \cdot \tan\left(\frac{\boldsymbol{x}}{2}\right)^{\odot 2} \right],$$

where $E^{(F)}$ is symmetric with zeros on its diagonal and $E^{(G)}$ is a strictly upper triangular matrix. This extended model has $2n^2 + 1$ degrees of freedom, which matches the number of evaluations exactly.

While the QAD model reconstructs the univariate restrictions of E to the coordinate axes correctly, the extended model $\mathring{E}$ does so for the bivariate restrictions to the plane spanned by any pair of coordinate axes. It remains to investigate whether and for which applications the extension yields a better optimization behaviour; for functions in which pairs of parameters yield a good local approximation of the landscape, it might provide an improvement.

The third modification we consider is to generalize the previous, extended QAD model to *general* single-parameter quantum gates. This can be done via a full trigonometric interpolation to second order, which is detailed in App. D.2, exactly reconstructing the energy function when restricted to any coordinate plane at the price of $2(\|\boldsymbol{R}\|_1^2 - \|\boldsymbol{R}\|_2^2 + \|\boldsymbol{R}\|_1) + 1$ evaluations.

Using toy model circuits and Hamiltonians, we demonstrate the qualitative difference between the QAD model, its extension $\mathring{E}$, and the generalization to multiple frequencies in Fig. 4.

6   Discussion

In this work, we derive interpolation rules to exactly express quantum functions $E(x)$ as a linear combination of evaluations $E(x_\mu)$, assuming $E(x)$ derives from parametrized gates of the form $U(x) = \exp(ixG)$. Our method relies on the observation that $E(x)$ can be expressed as trigonometric polynomial in x, characterized by a set of R *frequencies* that correspond to distinct differences in the eigenvalues of G. This observation allows us to derive our results using trigonometric interpolation methods.

[14]In addition, we may skip the n evaluations with shift angle π proposed in Ref. [49], and instead measure the Hessian diagonal as discussed in Sec. 4.1.
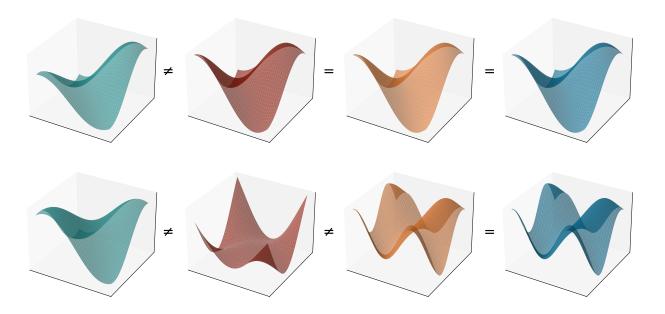
Figure 4: The QAD model (*left*), its extension $\mathring{E}$, see Eq. (53), that includes full second-order terms (*center left*), and the second-order trigonometric interpolation model (*center right*), as well as the original expectation value E (*right*). The original function is generated from toy Hamiltonians in a two-parameter example circuit, with one frequency (*top*) and two frequencies (*bottom*) per parameter. The QAD model produces a local approximation to E that deviates away from x_0 at a slow rate for $R = 1$ but faster for $R = 2$. The extension $\mathring{E}$ reuses evaluations made for the Hessian to capture the full bivariate dependence for a single frequency but is not apt to model multiple frequencies either. Finally, the trigonometric interpolation generalizes $\mathring{E}$. This means it coincides with $\mathring{E}$ for $R = 1$, but also reproduces the full bivariate function for $R > 1$.

In addition to a full reconstruction of $E(x)$, the presented approach offers parameter-shift rules for derivatives of arbitrary order and recipes to evaluate multivariate derivatives more cheaply. Using the concept of the stochastic parameter-shift rule, quantum gates of the form $U_F(x) = \exp(i(xG + F))$ can be differentiated as well.

Nevertheless, much remains unknown about the practicality of our new parameter-shift rules. For the common case that we have R equidistant frequencies, Sec. 3.5 shows that the scaling of the required resources is similar between naïvely applying our generalized parameter-shift rules, and applying parameter-shift rules to a decomposition of $U(x)$. This holds for the first derivative and also for the required shot budget when computing the second derivative, whereas the number of unique circuits is significantly smaller for the new, generalized shift rule.

Our observations lead to several open questions:

- In which situations can we obtain better bounds on the number of frequencies? We investigated an example for QAOA in Sec. 5.1, but are there other examples?

- For general G (e.g., $G = \sum_j c_j P_j$ with real c_j and Pauli words P_j), the frequencies will not be equidistant, and in fact R may scale quadratically in the size of U. Naïvely applied, our method would then scale poorly compared to decomposing G. Can we apply an approximate or stochastic parameter-shift rule with a better scaling?

- Would it ever make sense to *truncate* these parameter-shift rules to keep only terms corresponding to smaller frequencies? This is inspired by the idea of using low-pass filters to smooth out rapid changes of a signal.

- Our work on function reconstruction extends QAD to all gates with equidistant frequencies. Similarly, it allows Rotosolve, which has been shown to work remarkably well on some applications, to be used on all quantum gates with arbitrary frequencies. Is there a classification of problems on which these model-based algorithms work well? And can we reduce the optimization cost based on the above ideas?

- More generally, can we apply the machinery of Fourier analysis more broadly, e.g., to improve optimization methods in the presence of noise?

We hope that this work serves as an impetus for future work that will further apply signal processing methods to the burgeoning field of variational quantum computing.

Acknowledgements

Code availability

The scripts used to create the data and plots for Figs. 3 and 4 can be found at [79].

References

[1] Amazon Web Services. "Amazon Braket". url: aws.amazon.com/braket/.

[2] J.M. Arrazola, V. Bergholm, K. Brádler, T.R. Bromley, M.J. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, L.G. Helt, J. Hundal, T. Isacsson, R.B. Israel, J. Izaac, S. Jahangiri, R. Janik, N. Killoran, S.P. Kumar, J. Lavoie, A.E. Lita, D.H. Mahler, M. Menotti, B. Morrison, S.W. Nam, L. Neuhaus, H.Y. Qi, N. Quesada, A. Repingon, K.K. Sabapathy, M. Schuld, D. Su, J. Swinarton, A. Száva, K. Tan, P. Tan, V.D. Vaidya, Z. Vernon, Z. Zabaneh, and Y. Zhang. "Quantum circuits with many photons on a programmable nanophotonic chip". Nature 591, 54–60 (2021).

[3] IBM Corporation. "IBM Quantum". url: quantum-computing.ibm.com/.

[4] Microsoft. "Azure Quantum". url: azure.microsoft.com/../quantum/.

[5] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. "Parameterized quantum circuits as machine learning models". Quantum Science and Technology 4, 043001 (2019).

[6] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. "Variational quantum algorithms". Nature Reviews Physics 3, 625–644 (2021).

[7] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. "A variational eigenvalue solver on a photonic quantum processor". Nature Communications 5, 4213 (2014).

[8] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm" (2014). arXiv:1411.4028.

[9] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C. Benjamin. "Variational quantum algorithms for discovering Hamiltonian spectra". Phys. Rev. A 99, 062304 (2019).

[10] Gian-Luca R Anselmetti, David Wierichs, Christian Gogolin, and Robert M Parrish. "Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems". New Journal of Physics 23, 113010 (2021).

[11] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. "An adaptive variational algorithm for exact molecular simulations on a quantum computer". Nature communications 10, 1–9 (2019).

[12] Ken M. Nakanishi, Kosuke Mitarai, and Keisuke Fujii. "Subspace-search variational quantum eigensolver for excited states". Phys. Rev. Research 1, 033062 (2019).

[13] Alain Delgado, Juan Miguel Arrazola, Soran Jahangiri, Zeyue Niu, Josh Izaac, Chase Roberts, and Nathan Killoran. "Variational quantum algorithm for molecular geometry optimization". Phys. Rev. A 104, 052402 (2021).

[14] Eric Anschuetz, Jonathan Olson, Alán Aspuru-Guzik, and Yudong Cao. "Variational quantum factoring". In International Workshop on Quantum Technology and Optimization Problems. Pages 74–85. Springer (2019).

[15] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T. Sornborger, and Patrick J. Coles. "Quantum-assisted quantum compiling". Quantum 3, 140 (2019).

[16] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. "Hybrid quantum-classical approach to quantum optimal control". Phys. Rev. Lett. 118, 150503 (2017).

[17] Ryan LaRose, Arkin Tikku, Étude O'Neel-Judy, Lukasz Cincio, and Patrick J. Coles. "Variational quantum state diagonalization". npj Quantum Information 5, 1–10 (2019).

[18] Benjamin Commeau, Marco Cerezo, Zoë Holmes, Lukasz Cincio, Patrick J. Coles, and Andrew Sornborger. "Variational Hamiltonian diagonalization for dynamical quantum simulation" (2020). arXiv:2009.02559.

[19] Jonathan Romero, Jonathan P. Olson, and Alan Aspuru-Guzik. "Quantum autoencoders for efficient compression of quantum data". Quantum Science and Technology 2, 045001 (2017).

[20] Guillaume Verdon, Michael Broughton, and Jacob Biamonte. "A quantum algorithm to train neural networks using low-depth circuits" (2017). arXiv:1712.05304.

Accepted in ⟨ ⟩uantum 2022-03-18, click title to verify. Published under CC-BY 4.0.

16

[21] Edward Farhi and Hartmut Neven. "Classification with quantum neural networks on near term processors" (2018). arXiv:1802.06002.

[22] Maria Schuld and Nathan Killoran. "Quantum machine learning in feature Hilbert spaces". Phys. Rev. Lett. **122**, 040504 (2019).

[23] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. "Quantum circuit learning". Phys. Rev. A **98**, 032309 (2018).

[24] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. "Circuit-centric quantum classifiers". Phys. Rev. A **101**, 032308 (2020).

[25] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G. Green, and Simone Severini. "Hierarchical quantum classifiers". npj Quantum Information **4**, 1–8 (2018).

[26] Jin-Guo Liu and Lei Wang. "Differentiable learning of quantum circuit Born machines". Phys. Rev. A **98**, 062324 (2018).

[27] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. "Supervised learning with quantum-enhanced feature spaces". Nature **567**, 209–212 (2019).

[28] Hongxiang Chen, Leonard Wossnig, Simone Severini, Hartmut Neven, and Masoud Mohseni. "Universal discriminative quantum neural networks". Quantum Machine Intelligence **3**, 1–11 (2021).

[29] Nathan Killoran, Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. "Continuous-variable quantum neural networks". Phys. Rev. Research **1**, 033063 (2019).

[30] Gregory R. Steinbrecher, Jonathan P. Olson, Dirk Englund, and Jacques Carolan. "Quantum optical neural networks". npj Quantum Information **5**, 1–9 (2019).

[31] Andrea Mari, Thomas R. Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. "Transfer learning in hybrid classical-quantum neural networks". Quantum **4**, 340 (2020).

[32] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélémy Meynard-Piganeau, and Jens Eisert. "Stochastic gradient descent for hybrid quantum-classical optimization". Quantum **4**, 314 (2020).

[33] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: a system for large-scale machine learning". In OSDI. Volume 16, pages 265–283. Berkeley, CA, USA (2016). USENIX Association. url: dl.acm.org/..3026877.3026899.

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in PyTorch". NIPS 2017 Workshop Autodiff (2017). url: openreview.net/forum?id=BJJsrmfCZ.

[35] Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. "Autograd: Effortless gradients in NumPy". In ICML 2015 AutoML Workshop. (2015). url: indico.ijclab.in2p3.fr/..

[36] Atılım Güneş Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. "Automatic differentiation in machine learning: a survey". Journal of Machine Learning Research **18**, 1–153 (2018). url: http://jmlr.org/papers/v18/17-468.html.

[37] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M. Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, Keri McKiernan, Johannes Jakob Meyer, Zeyue Niu, Antal Száva, and Nathan Killoran. "PennyLane: Automatic differentiation of hybrid quantum-classical computations" (2020). arXiv:1811.04968.

[38] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. "Evaluating analytic gradients on quantum hardware". Phys. Rev. A **99**, 032331 (2019).

[39] Leonardo Banchi and Gavin E. Crooks. "Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule". Quantum **5**, 386 (2021).

[40] Gavin E. Crooks. "Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition" (2019). arXiv:1905.13311.

[41] Jakob S. Kottmann, Abhinav Anand, and Alán Aspuru-Guzik. "A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers". Chemical Science **12**, 3497–3508 (2021).

[42] Javier Gil Vidal and Dirk Oliver Theis. "Calculus on parameterized quantum circuits" (2018). arXiv:1812.06323.

[43] Francisco Javier Gil Vidal and Dirk Oliver Theis. "Input redundancy for parameterized quantum circuits". Frontiers in Physics **8**, 297 (2020).

[44] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. "Effect of data encoding on the expressive

power of variational quantum-machine-learning models". Phys. Rev. A **103**, 032430 (2021).

[45] Ken M. Nakanishi, Keisuke Fujii, and Synge Todo. "Sequential minimal optimization for quantum-classical hybrid algorithms". Phys. Rev. Research **2**, 043158 (2020).

[46] Andrea Mari, Thomas R. Bromley, and Nathan Killoran. "Estimating the gradient and higher-order derivatives on quantum hardware". Phys. Rev. A **103**, 012405 (2021).

[47] Johannes Jakob Meyer. "Fisher information in noisy intermediate-scale quantum applications". Quantum **5**, 539 (2021).

[48] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. "Quantum natural gradient". Quantum **4**, 269 (2020).

[49] Bálint Koczor and Simon C. Benjamin. "Quantum analytic descent" (2020). arXiv:2008.13774.

[50] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. "Structure optimization for parameterized quantum circuits". Quantum **5**, 391 (2021).

[51] Robert M. Parrish, Joseph T. Iosue, Asier Ozaeta, and Peter L. McMahon. "A Jacobi diagonalization and Anderson acceleration algorithm for variational quantum algorithm parameter optimization" (2019). arXiv:1904.03206.

[52] Artur F. Izmaylov, Robert A. Lang, and Tzu-Ching Yen. "Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations". Phys. Rev. A **104**, 062443 (2021).

[53] Oleksandr Kyriienko and Vincent E. Elfving. "Generalized quantum circuit differentiation rules". Phys. Rev. A **104**, 052417 (2021).

[54] Thomas Hubregtsen, Frederik Wilde, Shozab Qasim, and Jens Eisert. "Single-component gradient rules for variational quantum algorithms" (2021). arXiv:2106.01388v1.

[55] Antoni Zygmund. "Trigonometric series, Volume II". Cambridge University Press (1988).

[56] Kosuke Mitarai and Keisuke Fujii. "Methodology for replacing indirect measurements with direct measurements". Phys. Rev. Research **1**, 013006 (2019).

[57] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. "Variational ansatz-based quantum simulation of imaginary time evolution". npj Quantum Information **5** (2019).

[58] Ying Li and Simon C. Benjamin. "Efficient variational quantum simulator incorporating active error minimization". Phys. Rev. X **7**, 021050 (2017).

[59] David Wierichs, Christian Gogolin, and Michael Kastoryano. "Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer". Phys. Rev. Research **2**, 043246 (2020).

[60] Mauro E. S. Morales, Jacob D. Biamonte, and Zoltán Zimborás. "On the universality of the quantum approximate optimization algorithm". Quantum Information Processing **19**, 1–26 (2020).

[61] Seth Lloyd. "Quantum approximate optimization is computationally universal" (2018). arXiv:1812.11075.

[62] Matthew B. Hastings. "Classical and quantum bounded depth approximation algorithms" (2019). arXiv:1905.07047.

[63] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. "Quantum approximate optimization algorithm for MaxCut: A fermionic view". Phys. Rev. A **97**, 022304 (2018).

[64] Wen Wei Ho and Timothy H. Hsieh. "Efficient variational simulation of non-trivial quantum states". SciPost Phys **6**, 29 (2019).

[65] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices". Phys. Rev. X **10**, 021067 (2020).

[66] Matthew P. Harrigan, Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, et al. "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor". Nature Physics **17**, 332–336 (2021).

[67] Charles Delorme and Svatopluk Poljak. "The performance of an eigenvalue bound on the Max-Cut problem in some classes of graphs". Discrete Mathematics **111**, 145–156 (1993).

[68] William N. Anderson Jr. and Thomas D. Morley. "Eigenvalues of the Laplacian of a graph". Linear and Multilinear Algebra **18**, 141–145 (1985).

[69] Vladimir Brankov, Pierre Hansen, and Dragan Stevanović. "Automated conjectures on upper bounds for the largest Laplacian eigenvalue of graphs". Linear Algebra and its Applications **414**, 407–424 (2006).

[70] Michel X. Goemans and David P. Williamson. "Improved approximation algorithms for Maximum Cut and satisfiability problems using semidefinite programming". J. ACM **42**, 1115–1145 (1995).

[71] Miguel F. Anjos and Henry Wolkowicz. "Geometry of semidefinite MaxCut relaxations via matrix ranks". Journal of Combinatorial Optimization 6, 237–270 (2002).

[72] Liu Hongwei, Sanyang Liu, and Fengmin Xu. "A tight semidefinite relaxation of the MaxCut problem". J. Comb. Optim. 7, 237–245 (2003).

[73] Andrea Skolik, Jarrod R. McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. "Layerwise learning for quantum neural networks". Quantum Machine Intelligence 3, 1–11 (2021).

[74] Marcello Benedetti, Mattia Fiorentini, and Michael Lubasch. "Hardware-efficient variational quantum algorithms for time evolution". Phys. Rev. Research 3, 033083 (2021).

[75] Ernesto Campos, Aly Nasrallah, and Jacob Biamonte. "Abrupt transitions in variational quantum circuit training". Phys. Rev. A 103, 032607 (2021).

[76] Aharon Ben-Tal and Arkadi Nemirovski. "Lectures on modern convex optimization: Analysis, algorithms, and engineering applications". SIAM (2001).

[77] Elies Gil-Fuster and David Wierichs. "Quantum analytic descent (demo)". url: pennylane.ai/qml/demos/.. (accessed: 2022-01-23).

[78] Bálint Koczor (2021). code: balintkoczor/quantum-analytic-descent.

[79] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin (2022). code: dwierichs/General-Parameter-Shift-Rules.

[80] Leonard Benjamin William Jolley. "Summation of series". Dover Publications (1961).

[81] falagar. "Prove that $\sum_{k=1}^{n-1} \tan^2 \frac{k\pi}{2n} = \frac{(n-1)(2n-1)}{3}$". url: math.stackexchange.com/q/2343. (accessed: 2022-01-23).

A   Technical derivations

A.1   Derivation of explicit parameter-shift rules

Here we derive the trigonometric interpolation via Dirichlet kernels.

A.1.1   Full reconstruction

We start out by exactly determining $E(x)$ given its value at points $\{x_\mu = \frac{2\mu}{2R+1}\pi\}, \mu \in \{-R, \cdots, R\}$. This is a well-known problem [55, Chapter X]; we reproduce the result below for completeness.

Consider the *Dirichlet kernel*

$$D(x) = \frac{1}{2R+1} + \frac{2}{2R+1} \sum_{\ell=1}^{R} \cos(\ell x) \tag{54}$$

$$= \frac{\sin\left(\frac{2R+1}{2}x\right)}{(2R+1)\sin\left(\frac{1}{2}x\right)} \tag{55}$$

where the limit $x \to 0$ is taken when evaluating $D(0)$. The functions $D(x - x_\mu)$ are linear combinations of the basis functions $\{\sin(\ell x)\}_{\ell \in [R]}$, $\{\cos(\ell x)\}_{\ell \in [R]_0}$, and they satisfy $D(x_{\mu'} - x_\mu) = \delta_{\mu\mu'}$. Therefore it is evident that

$$E(x) = \sum_{\mu=-R}^{R} E(x_\mu) D(x - x_\mu) \tag{56}$$

$$= \frac{\sin\left(\frac{2R+1}{2}x\right)}{2R+1} \sum_{\mu=-R}^{R} E(x_\mu) \frac{(-1)^\mu}{\sin\left(\frac{x-x_\mu}{2}\right)}. \tag{57}$$

As an example, for $R = 1$ (e.g., when the generator G satisfies $G^2 = \mathbb{1}$) we have the formula

$$E(x) = \frac{\sin\left(\frac{3}{2}x\right)}{3} \left[-\frac{E(-\frac{2}{3}\pi)}{\sin(\frac{x}{2} + \frac{\pi}{3})} + \frac{E(0)}{\sin(\frac{x}{2})} - \frac{E(\frac{2}{3}\pi)}{\sin(\frac{x}{2} - \frac{\pi}{3})} \right]. \tag{58}$$

Derivatives of $E(x)$ can be straightforwardly extracted from this full reconstruction.

A.1.2   Odd kernels

We now consider the case of determining E_{odd} given its value at evenly spaced points $\{x_\mu = \frac{2\mu-1}{2R}\pi\}_{\mu \in [R]}$ [15]. Consider the *modified Dirichlet kernel*:

$$D^*(x) = \frac{1}{2R} + \frac{1}{2R} \cos(Rx) + \frac{1}{R} \sum_{\ell=1}^{R-1} \cos(\ell x) \tag{59}$$

$$= \frac{\sin(Rx)}{2R \tan\left(\frac{1}{2}x\right)} \tag{60}$$

where we again assume the limit $x \to 0$ is taken when evaluating $D^*(0)$. This kernel satisfies the relations

$$D^*(x_{\mu'} - x_\mu) = \delta_{\mu\mu'}, \quad D^*(x_{\mu'} + x_\mu) = 0, \tag{61}$$

but unfortunately, $D^*(x)$ is a linear combination of cosines, not sines; it's an even function, not an odd function. We therefore instead consider the linear combinations

$$\tilde{D}_\mu(x) := D^*(x - x_\mu) - D^*(x + x_\mu) \tag{62}$$

$$= \frac{\sin(R(x - x_\mu))}{2R \tan\left(\frac{1}{2}(x - x_\mu)\right)} - \frac{\sin(R(x + x_\mu))}{2R \tan\left(\frac{1}{2}(x + x_\mu)\right)}$$

$$= \frac{1}{R} \cos(x_\mu) \left[\frac{1}{2} \sin(Rx) + \sum_{\ell=1}^{R-1} \sin(\ell x) \right].$$

---

[15]Unlike Sec. A.1.1, we are not aware of a prior reference for the derivations for this subsection (reconstructing the odd part) and the next (reconstructing the even part).

Similarly to D^*, this kernel satisfies $\tilde{D}_\mu(x_{\mu'}) = \delta_{\mu\mu'}$ but it's a linear combination of the odd basis functions $\sin(\ell x), \ell \in [R]$. Following from these two properties, we know that

$$E_{\text{odd}}(x) = \sum_{\mu=1}^{R} E_{\text{odd}}(x_\mu) \tilde{D}_\mu(x) \tag{63}$$

$$= \sum_{\mu=1}^{R} \frac{E_{\text{odd}}(x_\mu)}{2R}$$

$$\times \left[\frac{\sin(R(x-x_\mu))}{\tan\left(\frac{1}{2}(x-x_\mu)\right)} - \frac{\sin(R(x+x_\mu))}{\tan\left(\frac{1}{2}(x+x_\mu)\right)} \right]$$

and we thus can reconstruct E_{odd} with the R evaluations $E_{\text{odd}}(x_\mu)$.

We also can extract from here a closed-form formula for the derivative at $x = 0$, as it only depends on the odd part of E. We arrive at the *general parameter-shift rule*:

$$E'(0) = \sum_{\mu=1}^{R} E_{\text{odd}}(x_\mu) \tilde{D}'_\mu(0) \tag{64}$$

$$= \sum_{\mu=1}^{R} E_{\text{odd}}(x_\mu) \frac{\sin(Rx_\mu)}{2R\sin^2(\frac{1}{2}x_\mu)} \tag{65}$$

$$= \sum_{\mu=1}^{R} E_{\text{odd}}\left(\frac{2\mu-1}{2R}\pi\right) \frac{(-1)^{\mu-1}}{2R\sin^2\left(\frac{2\mu-1}{4R}\pi\right)}.$$

Similarly, as the higher-order derivatives of $\tilde{D}_\mu$ can be computed analytically, we may obtain derivatives of E of higher odd orders.

A.1.3 Even kernels

Next we reconstruct the even part E_{even} again using the kernel $D^*(x)$ from above but choosing the $R+1$ points $x_\mu = \mu\pi/R$ for $\mu \in [R]_0$. As the spacing between these points is the same as between the previous $\{x_\mu\}$, we again have $D^*(x_{\mu'} - x_\mu) = \delta_{\mu\mu'}$; but note we cannot directly use $D^*(x - x_\mu)$ as our kernel because $D^*(x - x_\mu)$ is an even function in $x - x_\mu$ but not in x. Instead we take the even linear combination

$$\hat{D}_\mu(x) := \begin{cases} D^*(x) & \text{if } \mu = 0 \\ D^*(x - x_\mu) + D^*(x + x_\mu) & \text{if } 0 < \mu < R \\ D^*(x - \pi) & \text{if } \mu = R. \end{cases}$$

Then the $\hat{D}_\mu$ are even functions and satisfy $\hat{D}_\mu(x_{\mu'}) = \delta_{\mu\mu'}$, leading to

$$E_{\text{even}}(x) = \sum_{\mu=0}^{R} E_{\text{even}}(x_\mu) \hat{D}_\mu(x). \tag{66}$$

The second derivative of D^* is

$$D^{*\prime\prime}(x) = \frac{\sin(Rx)\left[1 - 2R^2\sin^2(\frac{1}{2}x)\right]}{4R\tan(\frac{1}{2}x)\sin^2(\frac{1}{2}x)} - \frac{\cos(Rx)}{2\sin^2\left(\frac{1}{2}x\right)}$$

and if we take the limit $x \to 0$:

$$D^{*\prime\prime}(0) = -\frac{2R^2+1}{6}. \tag{67}$$

This yields the explicit parameter-shift rule for the second derivative:

$$E''(0) = -E_{\text{even}}(0)\frac{2R^2+1}{6} + E_{\text{even}}(\pi)\frac{(-1)^{R-1}}{2}$$

$$+ \sum_{\mu=1}^{R-1} E_{\text{even}}\left(\frac{\mu\pi}{R}\right) \frac{(-1)^{\mu-1}}{\sin^2\left(\frac{\mu\pi}{2R}\right)}. \tag{68}$$

Again, derivatives of E of higher even order can be computed in a similar manner, using the same evaluations $E_{\text{even}}\left(\frac{\mu\pi}{R}\right)$.

A.2 Hessian parameter-shift rule

Here we consider the spectrum of the function

$$E^{(km)}(x) := E(\boldsymbol{x}_0 + x\boldsymbol{v}_{k,m}), \tag{69}$$

with $\boldsymbol{v}_{k,m} = \boldsymbol{v}_k + \boldsymbol{v}_m$. Without loss of generality, we assume U_k to act first within the circuit and set $\boldsymbol{x}_0 = \boldsymbol{0}$. As for the univariate case in Sec. 2.1, we may explicitly write the cost function as

$$E^{(km)}(x) = \langle\psi| U_k^\dagger(x)V^\dagger U_m^\dagger(x)BU_m(x)VU_k(x) |\psi\rangle$$

$$= \sum_{j_1,\dots j_4=1}^{d} \overline{\psi_{j_1}v_{j_2j_1}} b_{j_2j_3}v_{j_3j_4}\psi_{j_4} \tag{70}$$

$$\times \exp\left(i\left(\omega_{j_4}^{(k)} - \omega_{j_1}^{(k)} + \omega_{j_3}^{(m)} - \omega_{j_2}^{(m)}\right)x\right),$$

where $\omega^{(k,m)}$ are the eigenvalues of the generators of U_k and U_m, respectively, and we denoted the entries of matrices by lowercase letters as before. We may read off the occuring frequencies in this Fourier series in terms of the unique positive differences $\Omega^{(k,m)}$, leading to $\delta\Omega_{l_1l_2} = \pm\Omega_{l_1}^{(k)} \pm \Omega_{l_2}^{(m)}$. We again only collect the positive values as they come in pairs[16].

In case of integer-valued frequencies, there are $R_{km} = R_k + R_m$ such positive frequencies, namely all integers in $[R_k + R_m]$. For arbitrary frequencies, all $\{\delta\Omega\}$ might be unique and we obtain up to $R_{km} = 2R_kR_m + R_k + R_m$ frequencies. Rescaling the smallest frequency enforces a small degree of redundancy so that $R_{km} = 2R_kR_m + R_k + R_m - 2$ is always achievable; for some scenarios specific rescaling factors might drastically reduce R_{km} [17].

---

[16]That is, for any $\delta\Omega$, we also have $-\delta\Omega$ in the Fourier series, and the representation as real-valued function subsums the two frequencies.

[17]Recall that we used rescaling for the equidistant frequency case to arrive at integer-valued $\{\Omega\}$, which in turn made the significant reduction above possible.

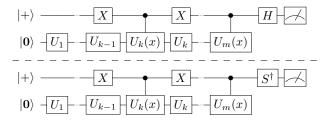Accepted in ⟨ ⟩uantum 2022-03-18, click title to verify. Published under CC-BY 4.0.

20

Figure 5: Circuits for the Hadamard tests to measure the overlap in Eq. (71), adapted from [57, Fig. 5]. The basis rotation in the last operation on the auxiliary qubit determines whether the real (*top*) or the imaginary (*bottom*) part of $\langle\psi(\boldsymbol{x}_0 + x\boldsymbol{v}_{k,m})|\psi(\boldsymbol{x}_0)\rangle$ is calculated. All unitaries without argument are understood as $U_j = U_j((\boldsymbol{x}_0)_j)$.

A.3 Hadamard tests for the metric tensor

In order to compute the metric tensor as the Hessian of the overlap $f(\boldsymbol{x}) = -\frac{1}{2}|\langle\psi(\boldsymbol{x})|\psi(\boldsymbol{x}_0)\rangle|^2$, we need to evaluate it at shifted positions $\boldsymbol{x} = \boldsymbol{x}_0 + x\boldsymbol{v}_{k,m}$. This can be done by executing the circuit $V(\boldsymbol{x}_0)$ and the adjoint circuit $V^\dagger(\boldsymbol{x})$ at the shifted position, and returning the probability to measure the $\boldsymbol{0}$ bitstring in the computational basis. As all operations after the latter of the two parametrized gates of interest cancel between the two circuits, those operations can be spared, but the maximal depth is (almost) the doubled depth of V.

Alternatively, we may use a Hadamard test as derived in the appendix of Ref. [57]. There, it was designed to realize the derivative overlaps $\mathfrak{Re}\{\langle\partial_k\psi(\boldsymbol{x})|\partial_m\psi(\boldsymbol{x})\rangle\}$ for the metric tensor directly, assuming the generator to be a Pauli word and therefore unitary. However, it can also be used to calculate the real or imaginary part of

$$
\begin{aligned}
\langle\psi(\boldsymbol{x})|\psi(\boldsymbol{x}_0)\rangle = \langle\boldsymbol{0}|\, U_1^\dagger((\boldsymbol{x}_0)_1)\cdots U_k^\dagger((\boldsymbol{x}_0)_k + x) \\
\cdots U_{m-1}^\dagger((\boldsymbol{x}_0)_{m-1})U_m^\dagger(x)U_{m-1}((\boldsymbol{x}_0)_{m-1}) \\
\cdots U_1((\boldsymbol{x}_0)_1)\,|\boldsymbol{0}\rangle\,.
\end{aligned}
\tag{71}
$$

by measuring the auxiliary qubit in the Z or Y basis. The corresponding circuit is shown in Fig. 5.

While the original proposal has to split up the generators into Pauli words and implement one circuit per combination of Pauli words from x_k and x_m, the number of circuits here is dictated by the number of evaluations in the parameter-shift rule. In order to measure $f(\boldsymbol{x})$, the real and the imaginary part both have to be measured, doubling the number of circuits.

A.4 Coefficient norms for univariate derivatives via equidistant shifts

The ℓ_1-norm of the coefficients in parameter-shift rules dictates the number of shots required to reach certain precision (see Sec. 2.3). Here, we explicitly compute this norm for both the general and decomposition-based parameter-shift rule for the first- and second-order univariate derivative. For the entire

analysis, we approximate the single-shot variance σ^2 to be constant as detailed in the main text.

A.4.1 Norm for general parameter-shift rule

For the case of equidistant shift angles, we can compute the norm of the coefficient vector $\boldsymbol{y}^{(1,2)}$ in the parameter-shift rules in Eqs. (24,25) explicitly, in order to estimate the required shot budget for the obtained derivative. For the first order, we note that the evaluations of E come in pairs, with the same coefficient up to a relative sign. This yields (recalling that $x_\mu = \frac{2\mu-1}{4R}\pi$):

$$
\|\boldsymbol{y}^{(1)}\|_1 = \frac{1}{2R}\sum_{\mu=1}^{R}\frac{1}{\sin^2(x_\mu)} = R,
\tag{72}
$$

which follows from $\sin^{-2}(x_\mu) = \cot^2(x_\mu) + 1$ and [80, Formula (445)]:

$$
\sum_{\mu=1}^{R}\cot^2(x_\mu) = 2R^2 - R.
\tag{73}
$$

A derivation for Eq. (73) can be adapted from Ref. [81], which we present below for completeness:

$$
\begin{aligned}
-i(-1)^\mu &= \exp(i2Rx_\mu) \\
&= \Big(\cos(x_\mu) + i\sin(x_\mu)\Big)^{2R} \\
&= \sum_{r=0}^{2R}\binom{2R}{r}(\cos(x_\mu))^{2R-r}(i\sin(x_\mu))^r \\
\Rightarrow\quad 0 &= \sum_{r=0}^{R}\binom{2R}{2r}(\cos(x_\mu))^{2R-2r}(i\sin(x_\mu))^{2r} \\
&= \sum_{r=0}^{R}\binom{2R}{2r}\Big(-\cot^2(x_\mu)\Big)^{R-r}
\end{aligned}
$$

Here we have applied the binomial theorem, extracted the real part, and divided by $(i\sin(x_\mu))^{2R}$ (note that $0 < x_\mu < \pi/2$). From the last equation above, we see that $\cot^2(x_\mu)$ is a root of the function $g(\chi) = \sum_{r=0}^{R}\binom{2R}{2r}(-\chi)^{R-r}$ for all $\mu \in [R]$. As g is a polynomial of degree R, we thus know *all* its roots and may use the simplest of Vieta's formulas:

$$
\sum_{\mu=1}^{R}\tau_\mu = -\frac{g_{R-1}}{g_R}
\tag{74}
$$

with roots $\{\tau_\mu\}_\mu$ of g, and g_j the jth order Taylor coefficient of g. Plugging in the known roots and coefficients we get

$$
\sum_{\mu=1}^{R}\cot^2(x_\mu) = -\frac{(-1)^{R-1}\binom{2R}{2}}{(-1)^R\binom{2R}{0}}
\tag{75}
$$

$$
= 2R^2 - R.
\tag{76}
$$

Accepted in ⟨ ⟩uantum 2022-03-18, click title to verify. Published under CC-BY 4.0.

21

For the second order we may repeat the above computation with small modifications[18], arriving at $g(\chi) = \sum_{r=0}^{R-1} \binom{2R}{2r+1}(-\chi)^{R-r}$ and therefore at

$$\|\boldsymbol{y}_1^{(2)}\| = \frac{2R^2+1}{6} + \frac{1}{2} + (R-1) - \frac{(-1)^{R-1}\binom{2R}{3}}{(-1)^R\binom{2R}{1}}$$
$$= R^2. \quad (77)$$

A.4.2   Norm for decomposition

If we compute the first- and second-order derivatives via a decomposition that contains $\mathcal{P}$ parametrized elementary gates, we need to apply the original two-term parameter-shift rule to each of these gates separately. For the first-order derivative, we simply sum all elementary derivatives. For integer-valued frequencies, x typically feeds without prefactor into the gates in the decomposition, so that the decomposition-based shift rule reads

$$E'(0) = \frac{1}{2\sin(x_1)} \sum_{k=1}^{\mathcal{P}} [E^{(k)}(x_1) - E^{(k)}(-x_1)], \quad (78)$$

where $E^{(k)}$ denotes the cost function based on the decomposition, in which only the parameter of the kth elementary gate is set to the shifted angle x_1 and to 0 in all other gates. To maximize $\sin(x_1)$, we choose $x_1 = \pi/2$, and as a reuslt all $2\mathcal{P}$ coefficients have magnitude $1/2$, and therefore

$$\|\boldsymbol{y}_{\text{decomp}}^{(1)}\|_1 = \mathcal{P}. \quad (79)$$

Due to all coefficients being equal, the optimal shot allocation is $N/(2\mathcal{P})$ for all terms.

For the second-order derivative, the full Hessian has to be computed from the decomposition as described in Ref. [46] and all elements have to be summed[19]:

$$E''(0) = \frac{1}{2\sin^2(x_1)} \sum_{\substack{k,m=1 \\ k<m}}^{\mathcal{P}} \quad (80)$$
$$\Bigg[E^{(km)}(x_1, x_1) - E^{(km)}(-x_1, x_1)$$
$$- E^{(km)}(x_1, -x_1) + E^{(km)}(-x_1, -x_1) \Bigg]$$
$$+ \frac{1}{2} \sum_{k=1}^{\mathcal{P}} [E^{(k)}(\pi) - E(0)]$$

where $E^{(km)}(x_1, x_2)$ is defined analogously to $E^{(k)}$ but the shift angles put into the kth and mth elementary gate may differ. Fixing the shift angle to $\pi/2$ again, we have $2\mathcal{P}(\mathcal{P}-1)$ coefficients of magnitude

---

[18]Recall that the angles differ between the two derivatives.

[19]Here we do not anticipate the cheaper Hessian evaluation from Sec. 4.1.

$1/2$ for the off-diagonal terms, $\mathcal{P}$ coefficients of magnitude $1/2$ for the $E^{(k)}(\pi)$ and one coefficient with magnitude $\mathcal{P}/2$ for $E(0)$, summing to

$$\|\boldsymbol{y}_{\text{decomp}}^{(1)}\|_1 = 2\mathcal{P}(\mathcal{P}-1)\frac{1}{2} + \mathcal{P}\frac{1}{2} + \frac{\mathcal{P}}{2} = \mathcal{P}^2. \quad (81)$$

Here the optimal shot allocation is to measure all shifted terms with $N/(2\mathcal{P}^2)$ shots, and $E(0)$ with $N/(2\mathcal{P})$ shots.

A.5   Coefficient norms for the Hessian

Similar to the previous section, we compute the coefficient norms for three methods to compute the Hessian for equidistant frequencies and shifts: We may use the diagonal shift rule in Eq. (36), repeat the general parameter-shift rule, or decompose the circuit and repeat the original parameter-shift rule. For the first approach, the diagonal entries of the Hessian— and thus the shifted evaluations for those entries—are reused to compute the off-diagonal ones, whereas the shifted evaluations for the repeated shift rule are distinct for all Hessian entries. This difference makes the cost comparison for a single Hessian entry difficult. We therefore consider the root mean square of the Frobenius norm of the difference between the true and the estimated Hessian as quality measure. The matrix of expected deviations is given by the standard deviations σ_{km} so that we need to compute

$$\varepsilon = \sqrt{\sum_{k,m=1}^{n} \sigma_{km}^2} = \sqrt{\sum_{k=1}^{n} \sigma_k^2 + \sum_{k<m} 2\sigma_{km}^2}. \quad (82)$$

A.5.1   Hessian shift rule

The variance for a Hessian diagonal entry H_{kk} is $\sigma^2 R_k^4 / N_{kk}$ if we use N_{kk} shots to estimate it (see Eq. (29))[20]. For an off-diagonal element H_{km} computed via the diagonal shift rule in Eq. (36), the variance is

$$\sigma_{km}^2 = \frac{1}{4}\left(\frac{\sigma^2(R_k+R_m)^4}{N_{km}} + \frac{\sigma^2 R_k^4}{N_{kk}} + \frac{\sigma^2 R_m^4}{N_{mm}} \right), \quad (83)$$

where we used that $R_{km} = R_k + R_m$ for equidistant frequencies. Overall, this yields

$$\varepsilon^2 = \sum_{k=1}^{n} \frac{\sigma^2 R_k^4}{N_{kk}}\frac{n+1}{2} + \sum_{k<m} \frac{\sigma^2(R_k+R_m)^4}{2N_{km}} \quad (84)$$

If we allocate N_{diag} shots optimally, that is N_{km} is proportional to the square root of the coefficient of N_{km}^{-1}, we require

$$N_{\text{diag}} = \frac{\sigma^2}{\varepsilon^2}\left[\sum_{k=1}^{n} R_k^2\sqrt{\frac{n+1}{2}} + \sum_{k<m} \frac{1}{\sqrt{2}}(R_k+R_m)^2 \right]^2$$
$$= \frac{\sigma^2}{2\varepsilon^2}\left[(\sqrt{n+1}+n-2)\|\boldsymbol{R}\|_2^2 + \|\boldsymbol{R}\|_1^2 \right]^2 \quad (85)$$

shots to estimate H to a precision of ε.

---

[20]Recall that σ^2 is the single-shot variance.

A.5.2 Repeated general parameter-shift rule

Without the diagonal shift rule, we compute H_{km} by executing the univariate general parameter-shift rule in Eq. (24) for x_k and x_m successively, i.e., we apply the rule for x_m to all terms from the rule for x_k. This leads to $4R_k R_m$ terms with their coefficients arising from the first-order shift rule coefficients by multiplying them together:

$$\|\boldsymbol{y}^{(km)}\|_1 = \frac{1}{4R_k R_m} \sum_{\mu=1}^{R_k} \frac{1}{\sin^2(x_\mu)} \sum_{\mu'=1}^{R_m} \frac{1}{\sin^2(x_{\mu'})}$$
$$= R_k R_m, \qquad (86)$$

where we used Eq. (72). Correspondingly, the variance for H_{km} computed by this methods with an optimal shot allocation of N_{km} shots is $\sigma_{km}^2 = \sigma^2 R_k^2 R_m^2 / N_{km}$. The mean square of the Frobenius norm then is

$$\varepsilon^2 = \sum_{k=1}^n \frac{\sigma^2 R_k^4}{N_{kk}} + \sum_{k<m} \frac{2\sigma^2 R_k^2 R_m^2}{N_{km}} \qquad (87)$$

and an optimal shot allocation across the entries of the Hessian to achieve a precision of ε will require

$$N_{\mathrm{genPS}} = \frac{\sigma^2}{\varepsilon^2} \left[\sum_{k=1}^n R_k^2 + \sum_{k<m} \sqrt{2} R_k R_m \right]^2$$
$$= \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{2} - 1)\|\boldsymbol{R}\|_2^2 + \|\boldsymbol{R}\|_1^2 \right]^2 \qquad (88)$$

shots in total.

A.5.3 Decomposition and repeated original shift rule

For the third approach, we only require the observation that again all (unique) Hessian entries are estimated independently and that the coefficients arise from all products of two coefficients from the separate shift rules for x_k and x_m. This yields $4\mathcal{P}_k \mathcal{P}_m$ coefficients with magnitude $1/4$, so that the calculation of ε is the same as for the previous approach, replacing $\boldsymbol{R}$ by $\mathcal{P}$. The required shot budget for a precision of ε is thus

$$N_{\mathrm{decomp}} = \frac{\sigma^2}{2\varepsilon^2} \left[(\sqrt{2} - 1)\|\boldsymbol{\mathcal{P}}\|_2^2 + \|\boldsymbol{\mathcal{P}}\|_1^2 \right]^2 \qquad (89)$$

B Generalization to arbitrary spectra

Throughout this work, we mostly focused on cost functions E with equidistant — and thus, by rescaling, integer-valued — frequencies $\{\Omega_\ell\}$. Here we will discuss the generalization to arbitrary frequencies, mostly considering the changed cost.

B.1 Univariate functions

The nonuniform DFT used to reconstruct the full function E in Sec. 3.1, and its modifications for the odd and even part in Secs. 3.2 and 3.3, can be used straightforwardly for arbitrary frequencies. However, choosing equidistant shift angles $\{x_\mu\}$ will no longer make the DFT uniform, as was the case for equidistant frequencies. Correspondingly, the explicit parameter-shift rules for $E'(0)$ and $E''(0)$ in Eqs. (24, 25) do not apply and in general we do not know a closed-form expression for the DFT or the parameter-shift rules. Symbolically, the parameter-shift rule takes the form

$$E'(0) = \sum_{\mu=1}^R y_\mu^{(1)} [E(x_\mu) - E(-x_\mu)] \qquad (90)$$

$$E''(0) = y_0^{(2)} E(0) + \sum_{\mu=1}^R y_\mu^{(2)} [E(x_\mu) + E(-x_\mu)]. \qquad (91)$$

Regarding the evaluation cost, the odd part and thus odd-order derivatives can be obtained at the same price of $2R$ evaluations of E as before, but the even part might no longer be periodic in general; as a consequence,

$$E_{\mathrm{even}}(\pi) = \frac{1}{2}(E(\pi) + E(-\pi)) \neq E(\pi) \qquad (92)$$

actually may require two evaluations of E, leading to $2R + 1$ evaluations overall. If the even part is periodic, which is equivalent to all involved frequencies being commensurable, with some period T, evaluating $E_{\mathrm{even}}(T/2)$ allows to skip the additional evaluation.

When comparing to the first derivative based on a decomposition into $\mathcal{P}$ parametrized elementary gates, the break-even point for the number of unique circuits remains at $R = \mathcal{P}$ as for equidistant frequencies, but we note that e.g., a decomposition of the form

$$U(x) = \prod_{k=1}^{\mathcal{P}} U_k(\beta_k x), \qquad (93)$$

namely where x is rescaled individually in each elementary gate by some $\beta_k \in \mathbb{R}$, in general will result in $R = \mathcal{P}^2$ frequencies of E, making the decomposition-based parameter-shift rule beneficial. For the second-order derivative, the number of evaluations $2R + 1$ might be quadratic in $\mathcal{P}$ in the same way, but the decomposition requires $2\mathcal{P}^2 - \mathcal{P} + 1$ as well, so that the requirements are similar if $R = \mathcal{P}$.

Regarding the required number of shots, we cannot make concrete statements for the general case as we don't have a closed-form expression for the coefficients $\boldsymbol{y}$, but note that for the decomposition approach, rescaling factors like the $\{\beta_k\}$ in Eq. (93) above have to be factored in via the chain rule, leading to a modified shot requirement.

An example for unitaries with non-equidistant frequencies would be the QAOA layer that implements the time evolution under the problem Hamiltonian (see Eq. (26)) for MaxCut on *weighted* graphs with non-integer weights.

For the stochastic parameter-shift rule in Sec. 3.6 we did not restrict ourselves to equidistant frequencies and derive it in App. C for general unitaries of the form $U_F = \exp(i(xG + F))$ directly.

B.2 Multivariate functions

While the univariate functions do not differ strongly for equidistant and arbitrary frequencies in E and mostly the expected relation between R and $\mathcal{P}$ changes, the shift rule for the Hessian and the metric tensor are affected heavily by generalizing the spectrum. First, the univariate restriction $E^{(km)}(x)$ in Eq. (34) still can be used to compute the off-diagonal entry H_{km} of the Hessian but this may require up to $2R_{km} + 1 = 4R_k R_m + 2R_k + 2R_m - 3$ evaluations (see App. A.2), in contrast to $2R_{km} = 2(R_k + R_m)$ in the equidistant case. Compared to the resource requirements of the decomposition-based approach, $4\mathcal{P}_k \mathcal{P}_m$, this makes our general parameter-shift rule more expensive if $R_k \gtrsim \mathcal{P}_k$.

As we use the same method to obtain the metric tensor $\mathcal{F}$, the number of evaluations grows in the same manner, making the decomposition-based shift rule more feasible for unitaries with non-equidistant frequencies. As $f(\boldsymbol{x}_0)$ does not have to be evaluated, an off-diagonal element $\mathcal{F}_{km}$ requires one evaluation fewer than H_{km}, namely $4R_k R_m + 2R_k + 2R_m - 4$.

C General stochastic shift rule

In this section we describe a stochastic variant of the general parameter-shift rule which follows immediately from combining the rule for single-parameter gates in Eq. (90) with the result from Ref. [39].

First, note that any shift rule

$$E'(x_0) = \sum_\mu y_\mu E(x_0 + x_\mu), \qquad (94)$$

with coefficients $\{y_\mu\}$ and shift angles $\{x_\mu\}$ for a unitary $U(x) = \exp(ixG)$, implies that we can implement the commutator with G:

$$i[G, \rho] = \sum_\mu y_\mu U(x_\mu)\rho U^\dagger(x_\mu), \qquad (95)$$

since the commutator between G and the Hamiltonian directly expresses the derivative of the expectation value $E'(0)$ on the operator level, and shift rules hold for arbitrary states.

Now consider the extension $U_F(x) = \exp(i(xG + F))$ of the above unitary. In the original stochastic

parameter-shift rule, the authors show[21]

$$E'(x_0) = \int_0^1 \mathrm{d}t \ \mathrm{tr}\left\{ U_F^\dagger(tx_0) B \, U_F(tx_0) \qquad (96) \right.$$

$$\left. \times i\left[G \ , \ U_F\big((1-t)x_0\big) |\psi\rangle\langle\psi| \, U_F^\dagger\big((1-t)x_0\big) \right] \right\}$$

where we again denoted the state prepared by the circuit before U_F by $|\psi\rangle$ and the observable transformed by the circuit following U_F by B. By using Eq. (95) to express the commutator, we obtain

$$E'(x_0) = \int_0^1 \mathrm{d}t \ \sum_\mu y_\mu \, \mathrm{tr}\left\{ U_F^\dagger(tx_0) B \, U_F(tx_0) \quad (97) \right.$$

$$\left. \times U(x_\mu) U_F\big((1-t)x_0\big) |\psi\rangle\langle\psi| \, U_F^\dagger\big((1-t)x_0\big) U^\dagger(x_\mu) \right\}.$$

We abbreviate the interleaved unitaries

$$U_{F,\mu}(x_0, t) := U_F(tx_0) U(x_\mu) U_F\big((1-t)x_0\big) \qquad (98)$$

and denote the cost function that uses $U_{F,\mu}(x_0, t)$ instead of $U_F(x_0)$ as

$$E_\mu(x_0, t) := \mathrm{tr}\left\{ B \, U_{F,\mu}^\dagger(x_0, t) |\psi\rangle\langle\psi| \, U_{F,\mu}(x_0, t) \right\}.$$

Rewriting Eq. (97) then yields the *generalized stochastic parameter-shift rule*

$$E'(x_0) = \int_0^1 \mathrm{d}t \sum_\mu y_\mu E_\mu(x_0, t). \qquad (99)$$

It can be implemented by sampling values for the splitting time t, combining the shifted energies $E_\mu(x_0, t)$ for each sampled t with the coefficients y_μ, and averaging over the results.

D Details on QAD

In this section we provide details on the latter two of the three modifications of the QAD algorithm discussed in Sec. 5.3.

D.1 Extended QAD model for Pauli rotations

The QAD model introduced in Ref. [49] contains trigonometric functions up to second (leading) order. The free parameters of the model cannot be extracted with one function evaluation per degree of freedom, because unlike standard monomials in a Taylor expansion, the trigonometric basis functions mix the orders in the input parameters. This leads to the mismatch of $2n^2 + n + 1$ (original QAD) or $3n^2/2 + n/2 + 1$ (see above) evaluations to obtain $n^2/2 + 3n/2 + 1$ model parameters. We note that the QAD model contains full univariate reconstructions at optimal cost, extracting

---

[21]To be precise, we here combine Eqs. (11-13) in Ref. [39] into a general expression for E'.

the $2n + 1$ model parameters $E^{(A)}$, $\boldsymbol{E}^{(B)}$ and $\boldsymbol{E}^{(C)}$ from $2n + 1$ function evaluations. The doubly shifted evaluations, however, are used for the Hessian entry only:

$$E_{km}^{(D)} = \frac{1}{4} \left[E_{km}^{++} - E_{km}^{+-} - E_{km}^{-+} + E_{km}^{--} \right], \quad (100)$$

where $E_{km}^{\pm\pm} = E(\boldsymbol{x}_0 \pm \frac{\pi}{2}\boldsymbol{v}_k \pm \frac{\pi}{2}\boldsymbol{v}_m)$ and we recall that this QAD model is restricted to Pauli rotations only.

Let us now consider a slightly larger truncation of the cost function than the one presented in App. A 2 in [49]:

$$\begin{aligned}
\mathring{E}(\boldsymbol{x}_0 + \boldsymbol{x}) = A(\boldsymbol{x}) &\Big[E^{(A)} \\
&+ 2\boldsymbol{E}^{(B)} \cdot \tan\left(\frac{\boldsymbol{x}}{2}\right) + 2\boldsymbol{E}^{(C)} \cdot \tan\left(\frac{\boldsymbol{x}}{2}\right)^{\odot 2} \\
&+ 4\tan\left(\frac{\boldsymbol{x}}{2}\right) E^{(D)} \tan\left(\frac{\boldsymbol{x}}{2}\right) \qquad (101) \\
&+ 4\tan\left(\frac{\boldsymbol{x}}{2}\right) E^{(F)} \tan^2\left(\frac{\boldsymbol{x}}{2}\right) \\
&+ 4\tan^2\left(\frac{\boldsymbol{x}}{2}\right) E^{(G)} \tan^2\left(\frac{\boldsymbol{x}}{2}\right) \Big]
\end{aligned}$$

with $A(\boldsymbol{x}) = \prod_k \cos^2(x_k/2)$. $E^{(F)}$ and $E^{(G)}$ have zeros on their diagonals because there are no terms of the form $\sin^3(x_k/2)$ or $\sin^4(x_k/2)$ in the cost function, and for $E^{(G)}$ we only require the strictly upper triangular entries due to symmetry. The higher-order terms contain at least three distinct variables x_k, x_l and x_m because all bivariate terms are captured in the above truncation. Using

$$A\left(\pm\frac{\pi}{4}\boldsymbol{v}_k \pm \frac{\pi}{4}\boldsymbol{v}_m\right) = \frac{1}{4} \quad \text{and} \quad \tan\left(\pm\frac{\pi}{4}\right) = \pm 1,$$

we now can compute:

$$\begin{aligned}
E_{km}^{++} - E_{km}^{-+} + E_{km}^{+-} - E_{km}^{--} &= E_k^{(B)} + E_{km}^{(F)} \\
E_{km}^{++} + E_{km}^{-+} + E_{km}^{+-} + E_{km}^{--} &= E^{(A)} + 2E_k^{(C)} \\
&\quad + 2E_m^{(C)} + 4E_{km}^{(G)}.
\end{aligned}$$

This means that the 4 function evaluations $E_{km}^{\pm\pm}$ that are used for $E_{km}^{(D)}$ in the original QAD can be recycled to obtain the 3 parameters $E_{km}^{(F)}$, $E_{mk}^{(F)}$ and $E_{km}^{(G)}$. The corresponding model is of the form Eq. (101) and therefore includes *all* terms that depend on two parameters only. Consequentially, the constructed model exactly reproduces the cost function not only on the coordinate axes but also on all coordinate planes spanned by any two of the axes. The number of model parameters is $2n^2 + 1$, which matches the total number of function evaluations.

D.2 Trigonometric interpolation for QAD

Both the original QAD algorithm, and the extension introduced above, assume the parametrized quantum circuit to consist of Pauli rotation gates exclusively. In the spirit of the generalized function reconstruction and parameter-shift rule, we would like to relax this assumption and generalize the QAD model. However, there is no obvious unique way to do this, because the correspondence between the gradient and $\boldsymbol{E}^{(B)}$ and between the Hessian and $\boldsymbol{E}^{(C,D)}$ is not preserved for multiple frequencies. Instead, the uni- and bivariate Fourier coefficients of E form the model parameters and the derivative quantities are contractions with the frequencies thereof. There are multiple ways in which we could generalize QAD to multiple frequencies.

The first way to generalize QAD is to compute the gradient and Hessian with the generalized parameter-shift rule Eq. (24) and the shift rule for Hessian entries Eq. (36) and to construct a single-frequency model as in original QAD. Even though we know the original energy function to contain multiple frequencies, this would yield a local model with the correct second-order expansion at $\boldsymbol{x}_0$ that exploits the evaluations savings shown in this work. As QAD is supposed to use the model only in the neighbourhood of $\boldsymbol{x}_0$, this might be sufficient for the optimization.

As a second generalization we propose a full trigonometric interpolation of E up to second order, similar to the univariate reconstruction in Sec. 3.1. First we consider the univariate part of the model: Start by evaluating E at positions shifted in the kth coordinate by equidistant points and subtract $E(\boldsymbol{x}_0)$,

$$E_\mu^{(k)} := E(\boldsymbol{x}_0 + x_\mu \boldsymbol{v}_k) - E(\boldsymbol{x}_0) \qquad (102)$$

$$x_\mu := \frac{2\mu\pi}{2R_k + 1}, \quad \mu \in [2R_k]. \qquad (103)$$

Then consider the (shifted) Dirichlet kernels

$$D_\mu^{(k)}(x) = \frac{1}{2R_k + 1} \left(1 + 2\sum_{\ell=1}^{R_k} \cos(\ell(x - x_\mu)) \right) \qquad (104)$$

$$= \frac{\sin\left(\frac{1}{2}(2R_k + 1)(x - x_\mu)\right)}{(2R_k + 1)\sin\left(\frac{1}{2}(x - x_\mu)\right)} \qquad (105)$$

which satisfy $D_\mu^{(k)}(x_{\mu'}) = \delta_{\mu\mu'}$ and are Fourier series with integer frequencies up to R_k. Therefore, the function[22]

$$\hat{E}^{(k)}(x) = \sum_{\mu=1}^{2R_k} E_\mu^{(k)} D_\mu^{(k)}(x) \qquad (106)$$

coincides with $E(\boldsymbol{x}_0 + x\boldsymbol{v}_k) - E(\boldsymbol{x}_0)$ at $2R_k + 1$ points and is a trigonometric polynomial with the same R_k frequencies.

---

[22]One might be wondering why to subtract $E(\boldsymbol{x}_0)$ just to add it manually back into the reconstruction now. This is because we need to avoid duplicating this term when adding up the univariate and bivariate terms of all parameters later on.

Similarly, the product kernels $D_{\mu\mu'}^{(km)}(x_k, x_m) = D_\mu^{(k)}(x_k)D_{\mu'}^{(m)}(x_m)$ can be used to reconstruct the bivariate restriction of E to the $x_k - x_m$ plane. For this, evaluate the function at doubly shifted positions and subtract both, $E(\boldsymbol{x}_0)$ and the univariate parts:

$$E_{\mu\mu'}^{(km)} := E(\boldsymbol{x}_0 + x_\mu \boldsymbol{v}_k + x_{\mu'} \boldsymbol{v}_m) \qquad (107)$$
$$- \hat{E}^{(k)}(x_\mu) - \hat{E}^{(m)}(x_{\mu'}) - E(\boldsymbol{x}_0) \qquad (108)$$

Then, the bivariate Fourier series

$$\hat{E}^{(km)}(x_k, x_m) = \sum_{\mu,\mu'=1}^{2R_k, 2R_m} E_{\mu\mu'}^{(km)} D_{\mu\mu'}^{(km)}(x_k, x_m) \qquad (109)$$

coincides with $E(\boldsymbol{x}_0 + x_k \boldsymbol{v}_k + x_m \boldsymbol{v}_m) - E(\boldsymbol{x}_0) - \hat{E}^{(k)}(x_k) - \hat{E}^{(m)}(x_m)$ on the entire coordinate plane spanned by $\boldsymbol{v}_k$ and $\boldsymbol{v}_m$.

As we constructed the terms such that they do not contain the respective lower order terms, we finally can combine them to the full trigonometric interpolation:

$$\hat{E}_{\text{interp}}(\boldsymbol{x}) = E(\boldsymbol{x}_0) + \sum_{k=1}^{n} \hat{E}^{(k)}(x_k) \qquad (110)$$
$$+ \sum_{k<m} \hat{E}^{(km)}(x_k, x_m).$$

This model has as many parameters as function evaluations, namely $2(\|\boldsymbol{R}\|_1^2 - \|\boldsymbol{R}\|_2^2 + \|\boldsymbol{R}\|_1) + 1$, and therefore, the trigonometric interpolation is the generalization of the extended QAD model in App. D.1. Indeed, for $R_k = 1$ for all k we get back $2(n^2 - n + n) + 1 = 2n^2 + 1$ evaluations and model parameters.

We note that the trigonometric interpolation can be implemented for non-equidistant evaluation points in a similar manner and with the same number of evaluations, although the elementary functions are no longer Dirichlet kernels but take the form

$$\mathring{D}_\mu^{(k)}(x) = \frac{\sin\left(\frac{1}{2}x\right)}{\sin\left(\frac{1}{2}x_\mu\right)} \prod_{\mu'=1}^{2R_k} \frac{\sin\left(\frac{1}{2}(x - x_{\mu'})\right)}{\sin\left(\frac{1}{2}(x_\mu - x_{\mu'})\right)}. \qquad (111)$$